

# Study and analysis on Optimal scheduling Data-Driven Peer-to-Peer Streaming

<sup>1</sup>B.Kundan <sup>2</sup>D.Jamuna

## Abstract:

We propose a new architecture for on-demand media streaming centered around the peer-to-peer (P2P) paradigm. The key idea of the architecture is that peers share some of their resources with the system. As peers contribute resources to the system, the overall system capacity increases and more clients can be served. In these applications, each node independently selects some other nodes as its neighbors (i.e., gossip style overlay construction) and exchanges streaming data with the neighbors (i.e., data scheduling). To improve the performance of such protocol, many existing works focus on the gossip-style overlay construction issue. However, few of them concentrate on optimizing the streaming data scheduling to maximize the throughput of a constructed overlay. In this paper, we analytically study the scheduling problem in data-driven streaming system and model it as a classical min-cost network flow problem. We then propose both the global optimal scheduling scheme and distributed heuristic algorithm to optimize the system throughput. Furthermore, we introduce layered video coding into data-driven protocol and extend our algorithm to deal with the end-host heterogeneity.

**Keywords:** Peer-to-peer systems, optimal scheduling, Distributed hash tables

## 1.INTRODUCTION:

A peer-to-peer, commonly abbreviated to P2P, distributed network architecture is composed of participants that make a portion of their resources (such as processing power, disk storage or network bandwidth) directly available to other network participants, without the need for central coordination instances (such as servers or stable hosts). Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model where only servers supply, and clients consume. Peer-to-peer was popularized by file sharing systems like Napster. Peer-to-peer file sharing networks have inspired new structures and philosophies in other areas of human interaction. In such social contexts, peer-to-peer as a meme refers to the egalitarian social networking that is

currently emerging throughout society, enabled by Internet technologies in general.

The basic idea of data-driven streaming protocol is very simple and similar to that of BitTorrent. The protocol contains two steps. In the first step, each node independently selects its neighbors so as to form an unstructured overlay network, called the gossip-style overlay construction or membership management. The second step is named block scheduling: The live media content is divided into blocks (or segments or packets), and every node announces what blocks it has to its neighbors. Then each node explicitly requests the blocks of interest from its neighbors according to their announcement. In this project, we present our analytical model and corresponding solutions to tackle the block scheduling problem in data-driven protocol. We first model this scheduling problem as a classical min-cost network flow problem and propose a global optimal solution in order to find out the ideal throughput improvement in theory. Since this solution is centralized and requires global knowledge, based on its basic idea, we then propose a heuristic algorithm that is fully distributed and asynchronous with only local information exchange. Furthermore, we employ layered video coding to encode the video into multiple rates and extend our algorithm to improve the satisfaction of the users with heterogeneous access bandwidth.

In Existing system we using two approaches for online media streaming called unicast based approaches and multicast based approach. There are two approaches that use unicast for on-demand streaming centralized content distribution networks (CDN). In the centralized approach we have maintaining the powerful server, this approach is easy to deploy and manage. But the scalability process leads to the failure in the network. There are two other critical, but less obvious, disadvantages of the centralized approach: high cost and load on the backbone network. On the negative side, this approach requires deploying and managing proxies at many locations. While deploying proxies increases the overall system capacity, it multiplies the cost. The capacity is still limited by the aggregate resources of the proxies. In multicast they have following two techniques

network level and application level in both they constructing the network in the tree and client act as a leaf.

Centralized approach: high cost and load on the backbone network. On the negative side, this approach requires deploying and managing proxies at many locations. While deploying proxies increases the overall system capacity, it multiplies the cost. The capacity is still limited by the aggregate resources of the proxies. In multicast they have following two techniques network level and application level in both they constructing the network in the tree and client act as a leaf.

The main purpose of the proposed system the scheduling problem in data-driven streaming system and model it as a classical min-cost network flow problem. We then propose both the global optimal scheduling scheme and distributed heuristic algorithm to optimize the system throughput. Furthermore, we introduce layered video coding into data-driven protocol and extend our algorithm to deal with the end-host heterogeneity. We present our analytical model and corresponding solutions to tackle the block scheduling problem in data-driven protocol. We first model this scheduling problem as a classical min-cost network flow problem and propose a global optimal solution in order to find out the ideal throughput improvement in theory. Since this solution is centralized and requires global knowledge, based on its basic idea, we then propose a heuristic algorithm that is fully distributed and asynchronous with only local information exchange. This scheduling problem as a classical min-cost network flow problem and propose a global optimal solution in order to find out the ideal throughput improvement in theory. Since this solution is centralized and requires global knowledge, based on its basic idea, we then propose a heuristic algorithm that is fully distributed and asynchronous with only local information exchange.

## 2. Peer-to-Peer systems:

A peer-to-peer, commonly abbreviated to P2P, distributed network architecture is composed of participants that make a portion of their resources (such as processing power, disk storage or network bandwidth) directly available to other network participants, without the need for central coordination instances (such as servers or stable hosts). Peer-to-peer networks are typically formed dynamically by ad-hoc additions of nodes. In an 'ad-hoc' network, the removal of nodes has no significant impact on the network. The distributed architecture of an application in a peer-to-peer system provides enhanced scalability and service robustness. Figure shows architecture of peer-to-peers.

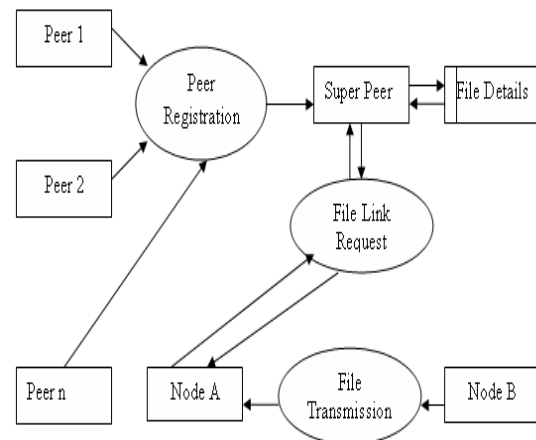


Fig: 2.1 Peer-to-peer systems Architecture

Peer-to-peer systems often implement an Application Layer overlay network on top of the native or physical network topology. Such overlays are used for indexing and peer discovery. Content is typically exchanged directly over the underlying Internet Protocol (IP) network. Anonymous peer-to-peer systems are an exception, and implement extra routing layers to obscure the identity of the source or destination of queries. In structured peer-to-peer networks, connections in the overlay are fixed. They typically use distributed hash table-based (DHT) indexing, such as in the Chord system (MIT). Unstructured peer-to-peer networks do not provide any algorithm for organization or optimization of network connections. In particular, three models of unstructured architecture are defined. In pure peer-to-peer systems the entire network consists solely of equipotent peers. There is only one routing layer, as there are no preferred nodes with any special infrastructure function. Hybrid peer-to-peer systems allow such infrastructure nodes to exist, often called supernodes. In centralized peer-to-peer systems, a central server is used for indexing functions and to bootstrap the entire system. Although this has similarities with a structured architecture, the connections between peers are not determined by any algorithm. The first prominent and popular peer-to-peer file sharing system, Napster, was an example of the centralized model. Gnutella and Freenet, on the other hand, are examples of the decentralized model. Kazaa is an example of the hybrid model. P2P networks are typically used for connecting nodes via largely ad hoc connections. Sharing content files (see file sharing) containing audio, video, data or anything in digital format is very common, and real

time data, such as telephony traffic, is also passed using P2P technology.

A pure P2P network does not have the notion of clients or servers but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client-server model where communication is usually to and from a central server. A typical example of a file transfer that is not P2P is an FTP server where the client and server programs are quite distinct: the clients initiate the download/uploads, and the servers react to and satisfy these requests. The P2P overlay network consists of all the participating peers as network nodes. There are links between any two nodes that know each other: i.e. if a participating peer knows the location of another peer in the P2P network, then there is a directed edge from the former node to the latter in the overlay network. Based on how the nodes in the overlay network are linked to each other, we can classify the P2P networks as unstructured or structured.

### **2.1. Structured peer-to-peer systems:**

Structured P2P network employ a globally consistent protocol to ensure that any node can efficiently route a search to some peer that has the desired file, even if the file is extremely rare. Such a guarantee necessitates a more structured pattern of overlay links. By far the most common type of structured P2P network is the distributed hash table (DHT), in which a variant of consistent hashing is used to assign ownership of each file to a particular peer, in a way analogous to a traditional hash table's assignment of each key to a particular array slot.

### **2.2. Unstructured peer-to-peer systems:**

An unstructured P2P network is formed when the overlay links are established arbitrarily. Such networks can be easily constructed as a new peer that wants to join the network can copy existing links of another node and then form its own links over time. In an unstructured P2P network, if a peer wants to find a desired piece of data in the network, the query has to be flooded through the network to find as many peers as possible that share the data. The main disadvantage with such networks is that the queries may not always be resolved. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful. Since there is no correlation between a peer and the content managed by

it, there is no guarantee that flooding will find a peer that has the desired data. Flooding also causes a high amount of signaling traffic in the network and hence such networks typically have very poor search efficiency. Most of the popular P2P networks are unstructured.

In pure P2P networks: Peers act as equals, merging the roles of clients and server. In such networks, there is no central server managing the network, neither is there a central router. Some examples of pure P2P Application Layer networks designed for file sharing are Gnutella and Freenet. There also exist hybrid P2P systems, which distribute their clients into two groups: client nodes and overlay nodes. Typically, each client is able to act according to the momentary need of the network and can become part of the respective overlay network used to coordinate the P2P structure. This division between normal and 'better' nodes is done in order to address the scaling problems on early pure P2P networks. Examples for such networks are for example Gnutella (after v0.4) or G2.

An other type of hybrid P2P network are networks using on the one hand central server(s) or bootstrapping mechanisms, on the other hand P2P for their data transfers. These networks are in general called 'centralized networks' because of their lack of ability to work without their central server(s). An example for such a network is the eDonkey network (eD2k).

### **2.3. Distributed hash tables:**

Distributed hash tables (DHTs) are a class of decentralized distributed systems that provide a lookup service similar to a hash table: (key, value) pairs are stored in the DHT, and any participating node can efficiently retrieve the value associated with a given key. Responsibility for maintaining the mapping from keys to values is distributed among the nodes, in such a way that a change in the set of participants causes a minimal amount of disruption. This allows DHTs to scale to extremely large numbers of nodes and to handle continual node arrivals, departures, and failures.

DHTs form an infrastructure that can be used to build peer-to-peer networks. DHT-based networks have been widely utilized for accomplishing efficient resource discovery for grid computing systems, as it aids in resource management and scheduling of applications. Resource discovery activities involve searching for the appropriate resource types that match the user's application requirements. Recent advances in the domain of decentralized resource discovery have been based on extending the existing DHTs with the

capability of multi-dimensional data organization and query routing. Majority of the efforts have looked at embedding spatial database indices such as the Space Filling Curves (SFCs) including the Hilbert curves, Z-curves, k-d tree, MX-CIF Quad tree and R\*-tree for managing, routing, and indexing of complex Grid resource query objects over DHT networks. Spatial indices are well suited for handling the complexity of Grid resource queries. Although some spatial indices can have issues as regards to routing load-balance in case of a skewed data set, all the spatial indices are more scalable in terms of the number of hops traversed and messages generated while searching and routing Grid resource queries.

#### **2.4. Advantages of P2P networks:**

In P2P networks, all clients provide resources, which may include bandwidth, storage space, and computing power. As nodes arrive and demand on the system increases, the total capacity of the system also increases. This is not true of a client-server architecture with a fixed set of servers, in which adding more clients could mean slower data transfer for all users. The distributed nature of P2P networks also increases robustness, and—in pure P2P systems—by enabling peers to find the data without relying on a centralized index server. In the latter case, there is no single point of failure in the system.

As with most network systems, insecure and unsigned codes may allow remote access to files on a victim's computer or even compromise the entire network. In the past this has happened for example to the Fast Track network when anti P2P companies managed to introduce faked chunks into downloads and downloaded files (mostly MP3 files) were unusable afterwards or even contained malicious code. Consequently, the P2P networks of today have seen an enormous increase of their security and file verification mechanisms. Modern hashing, chunk verification and different encryption methods have made most networks resistant to almost any type of attack, even when major parts of the respective network have been replaced by faked or nonfunctional hosts.

Internet service providers (ISPs) have been known to throttle P2P file-sharing traffic due to the high-bandwidth usage. Compared to Web browsing, e-mail or many other uses of the internet, where data is only transferred in short intervals and relative small quantities, P2P file-sharing often consists of relatively heavy bandwidth usage due to ongoing file transfers and swarm/network coordination packets. A possible solution to this is called P2P caching, where a ISP

stores the part of files most accessed by P2P clients in order to save access to the Internet.

### **3. Procedure:**

#### **3.1. Construction of peers:**

Peers in our system are organized in a way that facilitates keeping the traffic as local as possible. Peers are divided into clusters. A cluster is a logical grouping of peers that are topologically close to each other. Here we have maintaining the three parameters for the peer construction, and bandwidth allocation for the peers for file transfer, the maximum storage space the peer is willing to allocate to store segments of media files, the maximum number of concurrent connections that can be opened to serve requesting peers. One of the peers or a subset of them that seed new media files in the system. We choose the name seeding peers to indicate that their main functionality is to initiate the streaming service and not to serve all clients at all times.

#### **3.2. Creating the super peers and stream:**

Super peer is constructed to aid in locating the requested objects and disseminating the newly published ones. Super peers maintain information about the current peers in the system and the contents stored at each of them. Each super peer is responsible for a small fraction of the peers in the system. A stream is a time-ordered sequence of packets belonging to a specific media file. This sequence of packets is not necessarily downloaded from the same serving node. The packets should be downloaded before their scheduled display time to guarantee non-disruptive playback of the media.

#### **3.3. Initiating the streaming session:**

Here we are going to initiate the streaming session to share the available media files. For that we have three phases to initiate.

Phase-I Cluster-based searching algorithms:

Requesting peer checks for the availability of the desired media file in the system. This is done by sending a lookup request to the super peer to whom the requesting peer is attached. The super peer applies cluster-based searching algorithms and returns a list of candidate supplying peers.

Phase-II : Streaming by segment



It overlaps the streaming of one segment with the consumption of the previous segment. The playback of the media files starts after an initial buffering period.

### Phase-III : Media dispersion

The peer may cache some segments in order to disseminate the file into its cluster. Which segments a peer should cache is determined by the peer's level of cooperation and the dispersion algorithm

### 3.4. Peer maintenance:

Here we have to overcome the peer disconnection because peer are not reliable machines. To maintain full playback quality throughout the streaming session, a quality maintenance mechanism is needed. The quality maintenance mechanism has two parts: quality degradation detection and recovery.

### 3.5. Heuristic Distributed Algorithm: -

We give the algorithm to do global optimal scheduling. However, requiring global knowledge, such as block availability, bandwidth information, and request synchronization make the algorithm not scalable. Based on the basic idea of the global optimal solution, we present the heuristic practical algorithm, which is fully distributed and asynchronous. In our distributed algorithm, each node decides from which neighbor to fetch high blocks at the beginning of its request period. As the request period is relatively short, such as 3 seconds, our scheduling algorithm should make a decision as rapidly as possible. Therefore, in the distributed algorithm, we do a local optimal block scheduling on each node based on the current knowledge of the block availability among the neighbors. The local optimal block scheduling can also be modeled as a min-cost flow problem the sub-min-cost flow problem in the each rectangle is just the local optimal block scheduling.

## 4. Applications:

Active peer-to-peer technologies include:

- 1) Many file sharing networks, including Gnutella, G2 and Fast Track. Peer-to-peer files sharing popularized peer-to-peer technologies. As of 2009, it is the largest contributor of network traffic on the Internet.
- 2) Free, in depend internet in form of a Wireless community network or Netsukuku and Software publication and distribution (Linux, several games); via file sharing networks.

3) Research like the Chord project, the PAST storage utility, the P-Grid, and the system. and Distributed hash tables

4) In bioinformatics, drug candidate identification. The first such program was begun in 2001 the Centre for Computational Drug Discovery at the University of Oxford in cooperation with the National Foundation for Cancer Research. There are now several similar programs running under the United Devices Cancer Research Project. And the science net P2P search engine.

5) The U.S. Department of Defense has started research on P2P networks as part of its modern network warfare strategy. In May, 2003 Dr. Tether. Director of Defense Advanced Research Project Agency testified that U.S. Military is using P2P networks.

6) Delivery of TV content over a P2P network (P2PTV) Skype, one of the most widely used internet phone applications is using P2P technology. Osiris (Server less Portal System) allows its users to create anonymous and autonomous web portals distributed via P2P network.

7) VoIP (using application layer protocols such as SIP) and streaming media. P2PTV and PDTP. Applications include TV Player, Joost, Cool Streaming, Cyber sky-TV, TVants, PPLive, LiveStation

8)Peer casting for multicasting streams. See Peer Cast, Ice Share, FreeCast, Raw flow and Usenet for distributed discussion. See also list of Usenet newsreaders

Windows Peer-to-Peer. Distributed peer application development, collaboration Shipped with Advanced Networking Pack for Windows XP, Windows XP SP2, Windows Vista. This is a Windows component that runs only over IPv6 and provides a 'meta' peer-to-peer network that applications can utilize. It does not have file sharing support but third-parties can develop one. It also includes the Peer Name Resolution Protocol that allows dynamic domain name publication and resolution of names to endpoints. Windows Meeting Space and the People Near Me feature of Windows Vista use this protocol. It can be used to setup a Windows Internet Computer Name (WICN) using netsh p2p. Windows Remote Assistance and Home Group features of Windows 7 also use it. And cloud computing.

## 5. Conclusion:

In this paper, we study the scheduling problem in the data driven/swarming based protocol in peer-to-peer streaming. The contributions of this paper are two fold. First, to the best of our knowledge, we are the first to theoretically address the scheduling problem in data-driven protocol. Second, we give the optimal scheduling algorithm under different bandwidth constraints, as well as a distributed heuristic algorithm, which can be practically applied in real system and outperforms conventional ad hoc strategies by about 10 percent-70 percent in both single rate and multi rate scenario.

## References:

- [1] V. Agarwal and R. Rejaie, "Adaptive Multi-Source Streaming in Heterogeneous Peer-to-Peer Networks," Proc. Multimedia Computing and Networking (MMCN '05), Jan. 2005.
- [2] X. Zhang, J. Liu, B. Li, and T.-S.P. Yum, "Coolstreaming/Donet: A Data-Driven Overlay Network for Efficient Media Streaming," Proc. IEEE INFOCOM '05, Mar. 2005.
- [3] M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang, "A Peer-to-Peer Network for Live Media Streaming Using a Push-Pull Approach," Proc. ACM Multimedia, Nov. 2005.
- [4] N. Magharei and R. Rejaie, "Prime: Peer-to-Peer Receiver-Driven Mesh-Based Streaming," Proc. IEEE INFOCOM '07, May 2007 [5] Lin, E. T. and Delp, E. J.: "A Review of Data Hiding in Digital Images". Retrieved on 1.Dec.2006 from Computer Forensics, Cyber crime and Steganography Resources, Digital Watermarking Links and Whitepapers, Apr 1999.
- [5] Raja K B, C R Chowdary, Venugopal K R, L M Patnaik. (2005) : "A Secure Steganography using LSB, DCT and Compression Techniques on Raw Images," IEEE International Conference on Intelligence Sensing and Information processing, pp.171-176.
- [6] V. Agarwal and R. Rejaie, "Adaptive Multi-Source Streaming in Heterogeneous Peer-to-Peer Networks," Proc. Multimedia Computing and Networking (MMCN '05), Jan. 2005.

## Author's Information



<sup>1</sup> **B.Kundan** pursuing M.Tech CSE from jaya prakash Narayana College of Engineering & Technology B.Tech from jaya prakash narayana College of Engineering. Currently he is working as Assistant Professor at Jaya prakash Narayana College of Engineering. His areas of interest include Data mining, Network Security, Software Engineering and Sensor Network.



<sup>2</sup> **Prof.D.Jamunna**, Working as Professor & Head of CSE Dept. Jayaprakash Narayan College of Engineering, Mahabubnagar, M.Tech(SE) from School of Information Technology, JNTUH, Hyderabad. BE (CSE) from Vijayanagara Engineering College, Bellary. Experience 17 Years in Teaching Profession. Her areas of Interest are in Wireless Sensor Networks, Data Mining, and Networking and guided M. Tech and B. Tech Students IEEE Projects.