A Survey on Video Streaming Methods in Multihop Radio Networks

[1],R.RADHA, ASST.Prof. [2,] J U ARUN KUMAR
[1,2,]MRCET-HYD
[1,]MTECH(CSE)

Abstract

Multimedia ad multi hop networks are the prominent sources to deliver wide varieties of data. The data are in the form of motion with sound track called video, this type of data is differs from other formats and processing is alsodiffers. rocessing the video formatted data with compressed formats over network is streaming. The live video/realtime viewing applications work with streaming only.In this paper we mainly focuses on the video streaming in multihop networks with better approaches than fine grain and medium grain videos by concerning dynamic conditions . so we proposes the approaches to improve the video streaming with intellectual aspects of vedio quality and fairness among the vedio sessions, collision rates are to be considered in multihop radio networks.

Index Terms—Cross-layer optimization, dynamic spectrum access, distributed algorithm, multi-hop cognitive radio Networks.

1. Introduction

Stream processing is a computer programming paradigm, related to SIMD (single instruction, multiple data), that allows some applications to more easily exploit a limited form of parallel processing. Such applications can use multiple computational units, such as the FPUs on a GPU or field programmable gate arrays (FPGAs),[without explicitly managing allocation, synchronization, or communication among those units.The stream processing paradigm simplifies parallel software and hardware by restricting the parallel computation that can be performed. Given a set of data (a *stream*), a series of operations (*kernel functions*) is applied to each element in the stream. *Uniform streaming*, where one kernel function is applied to all elements in the stream, is typical. Kernel functions are usually pipelined, and local on-chip memory is reused to minimize external memory bandwidth. Since the kernel and stream abstractions expose data dependencies, compiler tools can fully automate and optimize on-chip management tasks. Stream processing hardware can use scoreboarding, for example, to launch DMAs at runtime, when dependencies become known. The elimination of manual DMA management reduces software complexity, and the elimination of hardware caches reduces the amount of the area not dedicated to computational units such as ALUs. Stream processing is essentially a compromise, driven by a data-centric model that works very well for traditional DSP or GPU-type applications (such as image, video and digital signal processing) but less so for general purpose processing with more randomized data access (such as databases). By sacrificing some flexibility in the model, the implications allow easier, faster and more efficient execution. Depending on the context, processor design may be tuned for maximum efficiency or a trade-off for flexibility.

Stream processing is especially suitable for applications that exhibit three application characteristics:[*citation needed*]

- **Compute Intensity**, the number of arithmetic operations per I/O or global memory reference. In many signal processing applications today it is well over 50:1 and increasing with algorithmic complexity.
- **Data Parallelism** exists in a kernel if the same function is applied to all records of an input stream and a number of records can be processed simultaneously without waiting for results from previous records.
- **Data Locality** is a specific type of temporal locality common in signal and media processing applications where data is produced once, read once or twice later in the application, and never read again. Intermediate streams passed between kernels as well as intermediate data within kernel functions can capture this locality directly using the stream processing programming model.

video streaming

When creating streaming video, there are two things you need to understand: The video *file format* and the *streaming* method.

File Formats

There are many video file formats to choose from when creating video streams. The most common formats are:

1. Windows Media
2. RealMedia
3. Quicktime
4. MPEG (in particular MPEG-4)
5. Adobe Flash

There are pros and cons for each format but in the end it comes down to personal preference. Be aware that many of your users will have their own preferences and some users will only use a particular format, so if you want to reach the widest possible audience you should create separate files for each format. In reality this isn't usually practical so you need to make a judgment call on which formats to provide. Obviously the better you understand all the options, the better your decision is likely to be.

Streaming Methods

There are two ways to view media on the internet (such as video, audio, animations, etc): *Downloading* and *streaming*.

Downloading

When you *download* a file the entire file is saved on your computer (usually in a temporary folder), which you then open and view. This has some advantages (such as quicker access to different parts of the file) but has the big disadvantage of having to wait for the whole file to download before any of it can be viewed. If the file is quite small this may not be too much of an inconvenience, but for large files and long presentations it can be very off-putting.The easiest way to provide downloadable video files is to use a simple hyperlink to the file. A slightly more advanced method is to *embed* the file in a web page using special HTML code.Delivering video files this way is known as HTTP streaming or HTTP delivery. *HTTP* means *Hyper Text Transfer Protocol*, and is the same protocol used to deliver web pages. For this reason it is easy to set up and use on almost any website, without requiring additional software or special hosting plans.

Note: This is not technically "true" video streaming — the best it can do is a passable imitation.

Streaming

Streaming media works a bit differently — the end user can start watching the file almost as soon as it begins downloading. In effect, the file is sent to the user in a (more or less) constant stream, and the user watches it as it arrives. The obvious advantage with this method is that no waiting is involved. Streaming media has additional advantages such as being able to broadcast live events (sometimes referred to as a *webcast* or *netcast*).True streaming video must be delivered from a specialized streaming server.

Progressive Downloading

There is also a hybrid method known as *progressive download*. In this method the video clip is downloaded but begins playing as soon as a portion of the file has been received. This simulates true streaming, but doesn't have all the advantages.

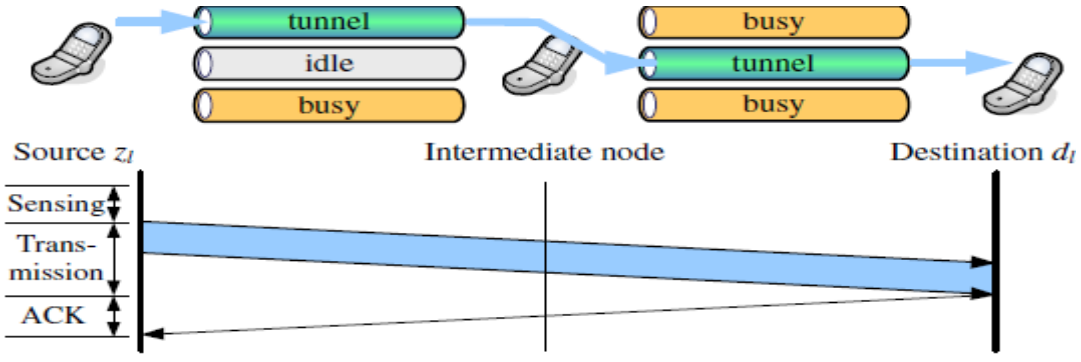


Fig. 2. The cut-through switching model for video data.

Compared to the traditional "hop-centric" approach, thisscheme greatly reduces the collision, contention, processing,and queuing delay induced at relay nodes . It is suitablefor real-time data with tight delay and jitter requirements.It is especially amicable for FGS video, since a corruptedpacket may still be useful for decoding. The viability, protocolrelatedissues, and practical considerations of this approach areaddressed in [8]. The challenging issue, however, is how toset up the tunnels, while the available channels at the relaysevolve over time due to primary user transmissions.

2. Streaming Video Servers

A streaming media or streaming video server is a specialized application which runs on an Internet server. This is often referred to as "true streaming", since other methods only simulate streaming. True streaming has advantages such as:

- The ability to handle much larger traffic loads.
- The ability to detect users' connection speeds and supply appropriate files automatically.
- The ability to broadcast live events.

There are two ways to have access to a streaming server:

1. Operate you own server (by purchasing or leasing)
2. Sign up for a hosted streaming plan with an ISP (Internet Service Provider)

Http Streaming Video

This is the simplest and cheapest way to stream video from a website. Small to medium-sized websites are more likely to use this method than the more expensivestreaming servers.For this method you don't need any special type of website or host — just a host server which recognises common video file types (most standard hosting accounts do this). You also need to know how to upload files and how to create hyperlinks (see our website tutorials for more info).

There are some limitations to bear in mind regarding HTTP streaming:

- HTTP streaming is a good option for websites with modest traffic, i.e. less than about a dozen people viewing at the same time. For heavier traffic a more serious streaming solution should be considered.
- You can't stream live video, since the HTTP method only works with complete files stored on the server.
- You can't automatically detect the end user's connection speed using HTTP. If you want to create different versions for different speeds, you need to create a separate file for each speed.
- HTTP streaming is not as efficient as other methods and will incur a heavier server load.

These things won't bother most website producers — it's normally only when you get into heavy traffic that you should be worried about them.

To Create HTTP Streaming Video

1. Create a video file in a common streaming media format
2. Upload the file to your web server
3. Make a simple hyperlink to the video file, or use special HTML tags to embed the video in a web page.

That's essentially all there is to it. When a user clicks the hyperlink, their media player opens and begins streaming the video file. If the file is embedded, it plays right there on the page.Now let's look at how to create the necessary video files...

Create a Streaming Video File

There are two ways to create stored streaming video files:

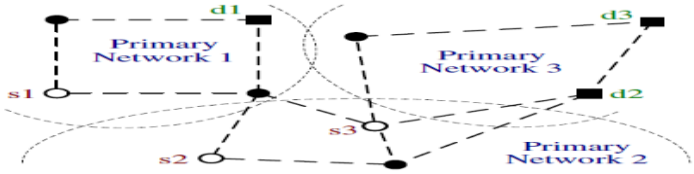
1. Use a conversion utility program. This takes an existing digital video file and converts it into the streaming format of your choice.
2. Export streaming files from video editing software such as Adobe Premiere, Final Cut Pro, etc.

Conversion Utilities

A conversion utility is a stand-alone program which imports a video clip and exports it to a different format. Examples include RealNetworksRealProducer and Sorenson SqueezeBasically, you simply open a file and select which which format to save it as. You can set various parameters to optimise the final video. The program will then chug away for some time while it makes the conversion.In the window shown here, the pane on the left shows the original file. The right pane shows the file as it is being converted to a streaming media format.

3. Considerations

The distributed algorithms are based on the fact that thecomputation is distributed on each feasible path. The OPTCSalgorithm requires information on channel availability andpacket loss rates at the links of feasible paths. The OPT-PSalgorithm computes the primal variable *yhl*for each path andbroadcasts Lagrangian multipliers over the control channel toall the source nodes.We assume a perfect control channel suchthat channel information can be effectively distributed andshared, which is not confined by the time slot structure [10].We assume relatively large timescales for the rimarynetwork time slots, and small to medium diameter for the CR network, such that there is sufficient time for timely feedbackof channel information to the video source nodes and forthe distributed algorithms to converge. Otherwise, channelinformation can be estimated using (5) based on delayedfeedback, leading to suboptimal solutions. If the time slot istoo short, the distributed algorithm may not converge to theoptimal solution



The most popular application of video streaming is VOD. VOD video files can be hosted by any server and it is possible to access them by anyone with a computer and some kind of Internet connection.The easy access may be extremely useful for all kinds of users, but especially for schools and universities, giving teachers an opportunity to archive classroom material. The lectures can then be accessed at any time by any student. What's more, they are easily supplemented with extra materials. Also, those universities which deal with distance learning rely mainly on video streaming technology.Generally, VOD is a very handy tool for any large institution that needs to distribute learning materials. Unlike in case of university TV, lectures can be pauses, rewound and reviewed, making all learning process much more effective. That's why sometimes educational videos are actually more effective than typical classroom instruction.Simply anything that can become a part of the footage can be kept on a web server and published on the Internet, including entertainment videos, business meetings, political speeches or 'virtual tours' which let the users see the layout of a particular building. Anther possible use is the security. Live video streaming can also help monitor remote locations. At first glance, this is similar to typical security cameras, except that streaming technology does not need closed circuits. The video can be monitored from virtually anywhere in the world, providing there is an Internet connection nearby. Streaming media technology – video streaming – enables the real time, or on-demand distribution of audio, video and multimedia on the internet. Streaming media is the simultaneous transfer of digital media (video, voice and data) so that it is received as a continuous real-time stream. Streamed data is transmitted by a server application and received and displayed in real-time by client applicationssuch as the Microsoft's Windows Media Player or the QuickTime Player. These applications can start displaying video or playing back audio as soon as enough data has been received and stored in the receiving station's buffer. A streamed file is simultaneously downloaded and viewed, but leaves behind no physical file on the viewer's computer.

4. Conclusion

We studied the problem of streaming multiple scalablevideos in a multi-hop CR network. The problem ormulationconsidered spectrum sensing and sensing errors, spectrumaccess and primary user protection, video quality and fairness, and channel/path selection for concurrent video sessions. Wefirst solved the formulated MINLP problem using a sequentialfixing scheme that produces lower and upper bounds on theachievable video quality. We then applied dual decompositionHU and Mao: Streaming Scalable Videos Over Multi-Hop Cognitive Radio Networks to derive a distributed algorithm, and analyzed its optimalityand convergence performance. Our simulations validated the efficacy of the proposed scheme.

References

1. Akyildiz, W. Lee, M. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: Asurvey," *Computer Netw. J.*, vol. 50, no. 9, pp. 2127-2159, Sept. 2006.
2. Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEESignal Process. Mag.*, vol. 24, no. 3, pp. 79-89, May 2007.
3. D. Hu, S. Mao, and J. Reed, "On video multicast in cognitive radionetworks," in *Proc. IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr.2009, pp. 2222-2230.
4. H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grainedscalable video coding method for multimedia streaming over IP," *IEEETrans. Multimedia*, vol. 3, no. 1, pp. 53-68, Mar. 2001.
5. M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC,"*IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1194-1203,
6. N. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wirelessnetworks: Efficient protocols and outage behavior," *IEEE Trans. Inf.Theory*, vol. 50, no. 11, pp. 3062-3080, Nov. 2004.
7. R. Kesavan and D. K. Panda, "Efficient multicast on irregular switchbasedcut-through networks with up-down routing," *IEEE Trans. ParallelDistrib. Syst.*, vol. 12, no. 8, pp. 808-828, Aug. 2001.
8. R. Ramanathan, "Challenges: A radically new architecture for next generationmobile ad-hoc networks," in *Proc. ACM MobiCom'05*, Cologne,Germany, Sept. 2005, pp. 132-139.
9. J. Jia, Q. Zhang, and X. Shen, "HC-MAC: A hardware-constrainedcognitive MAC for efficient spectrum management," *IEEE J. Sel. AreasCommun.*, vol. 26, no. 1, pp. 106-117, Jan. 2008.
10. H. Su and X. Zhang, "Cross-layer based opportunistic MAC protocolsfor QoSprovisionings over cognitive radio wireless networks," *IEEE J.Sel. Areas Commun.*, vol. 26, no. 1, pp. 118-129, Jan. 2008.
11. Motamedi and A. Bahai, "MAC protocol design for spectrumagilewireless networks: Stochastic control approach," in *Proc. IEEE DySPAN'07*, Dublin, Ireland, Apr. 2007, pp. 448-451.
12. H. Mahmoud, T. Yücek, and H. Arslan, "OFDM for cognitive radio:Merits and challenges," *IEEE Wireless Commun.*, vol. 16, no. 2, pp. 6-14, Apr. 2009.
13. Q. Zhao, S. Geirhofer, L. Tong, and B. Sadler, "Opportunistic spectrumaccess via periodic channel sensing," *IEEE Trans. Signal Process.*,vol. 36, no. 2, pp. 785-796, Feb. 2008.
14. M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptivecross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10,pp. 1752-1763, Dec. 2003.
15. Y. Hou, Y. Shi, and H. Sherali, "Spectrum sharing for multi-hopnetworking with cognitive radios," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 146-155, Jan. 2008.
16. http://en.wikipedia.org/wiki/Stream_processing, mediacollege.

1. Ms. R.Radha

B.Tech, M.Tech

Asst. Professor, CSE

2. J U ARUN KUMAR

B.C.A, M.C.A, M.Tech(Cse)

Mrcet, Maisammaguda

Dhulapally,Secunderabad