

MODELLING AND PERFORMANCE ANALYSIS FOR VIDEO ON DEMAND PRIOR STORING SERVER

Tony Tsang and Ho Fong Fun

Centre of International Education, Hong Kong College of Technology, Hong Kong

ABSTRACT

To handle the demand for mobile video is increased in the future. The mobile carriers should improve Video-on-Demand (VoD) System in order to increase video data transfer rate, then improve the system performance and quality of service (QoS). In this project, we suggest a new architecture for online video content in 4G LTE networks. The proposed structure includes a prior storing server in the eNode Base Station. It applies in every mobile network unit and mobile phone. In this work, we suggest a partial prior storing strategy. The Prior Storage Server is divided into Prior Storage 1 and Prior Storage 2. The video content depend on the popularity and value of utility to determine to store in Prior Storage 1 or Prior Storage 2. The partial prior storage can avoid eject the popular video content by the unpopular video content. In the project, we use RT-PEPA to model the behaviour of LTE Network and analyse the performance of LTE Network. Simulation results show that our proposed architecture can improve the system performance and QoS parameters in terms of packet loss, packet delay and jitter.

KEYWORDS

Video-on-Demand System, Wireless Network, Mobile Network, QoS parameters

1. INTRODUCTION

Long Term Evolution (LTE) provides wireless communication network of high speed data transmission between mobile phones and data terminals [1,2]. It base on UMTS/HSPA and GSM/EDGE network technologies. The LTE provide higher speed of wireless data networks and higher capacity compared with 3G Network. Although LTE provide fast network to us nowadays, mobile traffic will rise significantly in the future, it may cause network congestion which affect quality of user experience. There-fore, mobile network carriers must increase investment in network capacity (eg. transmission equipment, software and hardware).

Today, mobile video transmission is main source in mobile traffic and the request for mobile video will significantly grow in the future. Therefore, the mobile video content requires higher data transfer rate to handle increased mobile video transmission. The mobile carriers are necessary to improve video transmission in the mobile network.

To handle video transmission, the mobile carriers can improve the Video-on-Demand (VoD) system. A typical VoD system consists of many distributed serving nodes, connect via shared links. It builds up with many servers, and provides each individual piece of video on demand content to users. Conventional VoD system, e.g. IPTV, directly copy whole library of video content to all the serving nodes. It is insufficient with the rapidly extension of online video library. By comparison, collaborative VoD systems provide serving nodes to distribute their video content through in-system network, and become the popular systems nowadays.

In the area of media communication, video adaptation is a new technology. It degrades the video data transfer rate for users. It means that decrease quality of video, however the user can accept this downgraded quality of video and play the video more smoothly. As a result, It reduces the

network congestion. It aims to maximise the quality on experience (QoE) of all user equipment (UEs).

Another method to handle increased demand on Video-on-Demand (VoD) service, many geographically distributed serving nodes are constructed and joined via a high-bandwidth backbone. It consists of a lot of servers to save videos and gives Video-on-Demand service to users. We can distribute the videos between the nodes, and build them collaborative work. Thus, the volume of the system is significantly increases.

In this project, we aim to improve the performance of LTE Project, we propose a new architecture for LTE Network. Moreover, we suggest to divide prior storage server into Prior Storage 1 and Prior Storage 2 to store video content. Though this new architecture for LTE Network, We can improve QoS parameters and the system performance of LTE Network.

2. PROPOSED ARCHITECTURE

2.1 PRIOR STORAGE SERVERS IN LTE SYSTEM

To manage Video Content, a prior storing serve in the eNode Base Station communicate with Mobile Phone through LTE Network. In this structure, we suggest a partial prior storing strategy. It base on the popularity of the video content segments during the period and the access patterns to calculate the utility of the contents in the prior storage server. We also propose to partial prior storage to avoid replacing the popular contents by the unpopular contents. Simulation results show that our proposed structure can upgrade QoS parameters and improve the system performance. The distributed local video content servers allow operators to economically relieve network bandwidth limitation and the inherent storage, proportionally distribute the video content and demand of user. In this work, an efficient prior storing management scheme is investigated for distributed IPTV local video content services. Video Content prior storage is a basic method for improving the quality of service and performance to consumer.

In our proposed architecture in Figure1, the ISP network (operated by a individual telecom supplier) for the VOD services in which the video contents are stored at the Video Content Server and connect to Video Content Distribution Networks and Packet Data Network Gateways. The multiple eNode Base Station are connected to Packet Data Network Gateways and connected to Prior Storage Server individually. We suppose that video contents are divided into different parts of content segments with constant size and the number of bytes played from content segment will record in order to calculate the value of utility of content segment. Therefore, it capture the utility of prior storing the contents segment. We suppose that different video contents parts of the same video have different popularity. Popularity of a video will change when the user find to see different parts of the video. Therefore, some video content segments have a higher popularity than other content segment. Then, we use a content segment as a basic unit of a video content for prior storing and replacement. Following, the basic unit of a video use a large piece payload and a content segment can be defined as a collection of payloads. On the other hand, we purpose to partition the prior store into two level prior storages. Each partitioned prior storing updates the utility of content segments in its storage. The user ratings will decide the popularity distribution of the video. Prior Storage 1 (primary prior storage) use to prior storage a content segment that access for the first time (on a typical prior storage miss) and Prior Storage 2 (secondary prior storage) use to store content segments whose utility is high (i.e., popular over a longer time). In mean that the utility of content segments will decide the video content is stored in Prior Storage 1 or Prior Storage 2. Finally, the multiple eNode Base Station connected to Prior Storage Server to save the data of video. On the other side, the eNode Base Station communicated with mobile phone. Therefore, the user can get the data of video. In summary, we proposed a dual level prior storage algorithm which adaptively prior storage with bi-level control and it incorporated with popularity characteristic of online video streaming service. It is important that the Node Base

Station as a caching server to communicate with mobile phone to user. By our proposed architecture can upgrade QoS parameters and improve the LTE system performance.

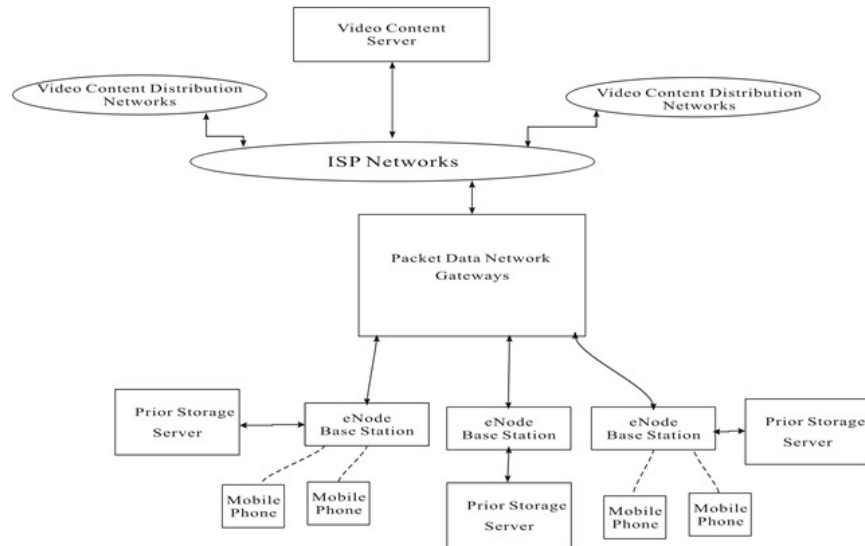


Figure1: Network Model for Prior Storage Servers in a LTE System

3. RT-PEPA AND ITS TOOLS

Network simulation is a technique where a program simulate the behavior of a network using mathematical formulas. The behavior of the network and the various performance parameter can analyse in a software; The network configuration/model reports the state of the network (links, routers, nodes and switches) and the events (packet error and data transmissions etc.)

A network simulator is tool that predicts the behavior of a network. In simulators, the network is typically simulated with applications, devices and links etc. and the performance of the network is analysed. In the project, We use Real Time – Performance Evaluation Process Algebra (RT-PEPA) as a network simulation to evaluate the performance of LTE Network. It describes the real-time stochastic behavior of LTE Network. This methodology analyses the performance of network using an automatic tool. It integrates conventional stochastic process algebra with real time to describe complex system. It includes parallel composition, timed transition, hard-time aspects and probabilistic branching. It describes the behavior of System. It includes activities about an interaction of the components. Moreover, the process measure performance from the model. The PEPA [5] formalism provides different operators. The operators have components for individual activities. It mean that different operation have different function to model the network. In this section, we give a brief introduction of the operators.

4.1. NOTATION AND DEFINITIONS OF RT-PEPA

For easy reading in following section, so we show the notation and definition of RT-PEPA [6-8]: Real-Time - Performance Evaluation Process Algebra (RT-PEPA) is process algebraic language. It provides the compositional description of distributed systems and concurrent and analyses their performances. The fundamental elements of RT- PEPA are its activities and operators. Activities would simulate behavior of the systems and operators use to compose algebraic descriptions. Time point

A time point is a time instant with the system; it does not have duration. It indicates the time of start and stop of an activity. Actually, the time point mean that use a discrete time domain, which contains the following properties:

$$\forall t \exists t' < t \wedge \forall t'' : t < t'' \Rightarrow t' \leq t''$$

We suppose a fixed set of clocks $t = \{t_0, \dots, t_i\}$. The particular time point t_0 , It indicates the start time point, always has the value 0.

Time Constraint

The time constraint of the action is defined the duration when an action exists. A time constraint has a start and an end point. It include a lower-bound and an upper-bound time point. Actually, we define a time constraint in the following:

$$T_i = \{[t_{\min}, t_{\max}] \mid \forall t_i \in T\} \text{ with } 0 \leq t_{\min} \leq t_{\max} .$$

Timed Action

A timed action consisting 3 elements $\langle \alpha, \lambda, T \rangle$. Where α is type of the activity, λ is the rate of the activity and T is temporal constraint of the activity. The type means the kind of activity, such as transmission of data packets. The rates represent the random variables at the duration of the activities. The actions can define different probability distribution function such as Constant, Exponential, Poisson, Uniform distribution and Geometric. Furthermore, temporal constraint bind each transition. In the following, we introduce the operation semantics and basic notations about RT-PEPA. The syntax of RT-PEPA can represent following:

The operators of Performance Evaluation Process Algebra (PEPA) are introduced in the below:

- stop define as inactive process.
- $\langle \alpha, \lambda, T \rangle . P$, is a prefix operator. The type of the activity is a probability distribution function type α , λ denote the rate of activity and T is the temporal constraint of component. The next acts as P . Series of activities can combine to generate a time constraint for an activity.
- $P + Q$ is choice combinator. This component may act either as P or as Q . All the current activities P and Q are enabled. The first activities completely distinguish which processes to use. When the first activity is completed, the other process of the choice stop to operation. Then, the system will generate resulting from the development of the chosen process.
- $P \oplus_r T Q$ represents the probabilistic choice with the traditional generative explanation, thus with probability r the process acts like P and with probability $1 - r$ it acts like Q bounded with the time constraint T .

- $P \parallel_{L,T} Q$ is a cooperation. This component represents the two activities P and Q are parallel, L, T synchronize with all activities whose type is in the cooperation set L of activity types.

The lifetime of two activities is the time constraint T . When the time constraint expires, these two activities stop to operation. If any type of activities is not in L , the activities will proceed individually. The set L is cooperation set and the lifetime of two activities is T .

- $P \sum_{P,T} Q$ is a unary operator. This component returns the set of activities that meet the temporal predicate condition. It consist of the boolean connectives: 'Or', 'And', 'Not' and Exclusive-Or (EXOR)'and

- P/L is a hiding operation. In this component, the set L of visible activities types specifies that those activities consider internal or private to the component.

- $A := P$ is a countable set of constants.

3.2. RT-PEPA ECLIPSE PLUG-IN TOOLS

The PEPA Eclipse Plug-in Tools is a software. It supports the stochastic process algebra PEPA. The PEPA is a formal language for modelling systems. It allows quantitative analysis of systems. It includes a number of interacting components run in parallel and their behavior is stochastic. PEPA language have Continuous Time Markov Chains (CTMCs) and Ordinary Differential Equations (ODEs). They have two consistent quantitative semantic interpretations and distinctive strength. PEPA can apply in wide variety of systems. In this project, we use a new addition to the PEPA plugin, The aggregating or combining states can abstract the model of PEPA plugin. To simulate the behavior of the network, we use two abstraction techniques - abstract stochastic bounds and CTMCs. The model check is (CSL) Continuous Stochastic Logic, It calculates the probability of a quantitative property about the abstraction a safe bound in the original model X if the real probability is p , then the model checker will return an interval. The software of the PEPA plug-in can download from

<http://www.dcs.ed.ac.uk/pepa/tools/plugin>, and provides several views:

Abstract Syntax Tree View

The Abstraction Syntax Tree View is a graphical interface. This view connects the active PEPA editor in the workspace. Moreover, It shows the state space of each sequential component about PEPA model. It also shows the abstract syntactic structure of PEPA model. It is useful for debugging purpose and the developers can easy to control PEPA abstract syntax trees programmatically.

Model Checking View

The Model Checking View is an interface. It can edit, construct and model checking CSL properties. The property editor can construct CSL formulae. In the abstraction view, the state gives labels of reference. It makes sure that only syntactically well-formed CSL formulae can construct.

State Space View

The State Space View connects to the PEPA editor and provides a tabular representation with state space of the underlying Markov chain. The state of the Markov chain show in a row and every cell shows the local state of a sequential component in the table.

We can manage state space filters in a toolbar menu item of user software interface. When a set of filter rules is turn on, the table removes the excluded states. The filters automatically calculate the probability in the state and show in the view.

On the other hand, the view has a Single-step Debugger. It can navigate the transition system of the Markov chain. The state describes the current state and two tables display in an external window. The tables show the set of states. Another, a further column shows the action types about transition. The user can navigate forwards and backwards by selecting the states.

Performance Evaluation View and Graph View

The user can perform steady-state analysis on the Markov chain. In the Performance Evaluation View, Performance metrics are computed automatically and showed. It has three tabs that show the results of the reward structures (utilization, population and throughput levels). Population and Throughput levels arrange in a tabular fashion, whereas utilization is shown in a two-level tree. The Performance Evaluation View can export input to the Graph View, it can show visualizing charts. Population and Throughputs levels are shown as bar charts. Another, pie chart shows in a

top-level node of the utilization tree. In the view, you can convert different kind of graph to display. The graph can export to SVG or PDF and the data can export into a separated text file.

Experimenting with Markovian Analysis

Sensitivity analysis is an important stage in performance modelling, i.e. the certain parameters can reflect performance of the system. The user can build up sensitivity analysis experiments in software. The performance metrics can be analysed by utilization, population or throughput levels. The software allows to build up multiple experiments of two kinds: one-dimensional or two-dimensional. The results of the analysis show in the Graph View as line charts, it can analyse performance of network.

Time Series Analysis

There are three basic steps to complete when executing a time-series analysis. Component selection, solver selection and solver parameterization are execute the time-series analysis. The software allows to select suitable components in the system. Parameterization and Solver selection are self-explanatory, with the parameters connect to the selected solver. Finally, the parameters, including selected solver will show the results in the graph view for our reference. The last characteristic of the wizard can export the model in another format, for example Matlab.

3.3. OVERALL SYSTEM SPECIFICATION

The basic LTE Network consists of four main parts. Internet Network, Base Station Sever and Mobile Phone. The Base Station connects the Internet Network first. The Base Station also connects to Prior Storage Server. In this paper, Prior Storage Server is useful for store video content. Prior Storage Server is divided into Prior Storage 1 and Prior Storage 2. When user requires data of video content, the mobile phone will connect to the Base Station through wireless communication medium. The Base Station will fetch the video content from Prior Storage Server and send the data of video content to user. The basic LTE Network show in below as figure 2:

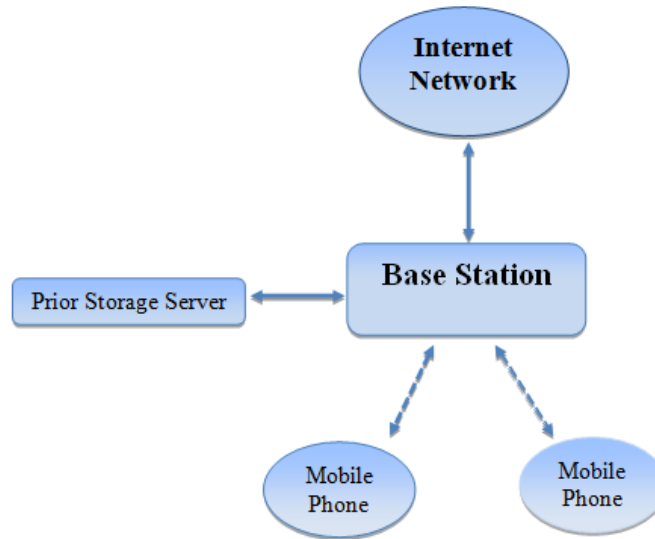


Figure 2: Basic Structure for LTE Network

In this section, we use CTMC-based analytical models using PEPA. We choose cooperation $P \boxtimes_{L,T} Q$ as a operator in PEPA in this test. This component represents the parallel operation in Network, it can directly reflects the Network structure. Then, since PEPA is a process algebra

language, it is quicker and easier to construct models than working directly at the state space level. Another, PEPA models can be solved numerically, some restrictions. Finally, PEPA component describes the behavior of the LTE network, the performance of the LTE network will analyse by the software.

Overall system Specification using RT-PEPA in the following:

$LTE\ Network := Internet\ net \xrightarrow{L,T} Base\ Station \xrightarrow{L,T} Prior\ Storage\ Sever \xrightarrow{L,T} Mobile\ Phone$

Prior storage server is divided into Prior Storage 1 and Prior storage 2. Video content depend on the value of utility and its popularity distribution to decide to enter Prior Storage 1 and Prior storage 2. The video content will effectively allocate in Prior storage server. In this structure, It can improve the performance of LTE network .

Prior storage server Specification using RT-PEPA in the following:

$Prior\ storage\ server := Prior\ Storage\ 1 \xrightarrow{L,T} Prior\ Storage\ 2$

In Prior storage 1 and Prior storage 2, α -search procedure and α -prior storing procedure is process when store video content. The α -prior storing procedure consists of the following steps.

- Step 1: Save top popular video content with α fraction of system size;
- Step 2: Save uncovered video content clips with rest size;
- Step 3: Fill the rest of disk.

Prior storage 1 specification using RT-PEPA in the following :

$Prior\ Storage\ 1 := \alpha\text{-prior\ storing\ procedure} \xrightarrow{L,T} \alpha\text{-search\ procedure}$

Prior storage 2 specification using RT-PEPA in the following:

$Prior\ storage\ 2 := \beta\text{-prior\ storing\ procedure} \xrightarrow{L,T} \beta\text{-search\ procedure}$

4. PERFORMANCE EVALUATION

The UDP Echo Plus protocol solves time synchronization problem and performs measurement to collect one-way QoS information. When a UDPEcho Plus capable service receives a standard UDPEcho packet, the UDPEchoPlus server just reflects the request back towards the source. Five UDP Echo Plus parameters are added to solve the packet loss and one-way jitter, and defined as following:

- **TESTGENSN:** The packet's sequence number of request packet set by UDP Echo Plus client
- **TESTRESPSN:** The UDP Echo Plus server's count that is incremented upon each valid echo request packet it received and responded to. This count is set to 0 when the UDP Echo server is enabled.

TESTRESPRECVTIMESTAMP: It is set by the UDP Echo Plus server to record the reception time of request packet.

- **TESTRESPREPLYTIMESTAMP:** It is set by the UDP Echo Plus server to record the forwarding time of response packet.

- **TESTRESPREPLYFAILURECOUNT:** It is set by the UDP Echo Plus server to record the number of locally dropped echo response packets. This count is incremented if a valid echo request packet is received but for some reason cannot be responded to (e.g. due to local buffer overflow, CPU utilization, etc.). It is a cumulative count with its current value placed in all request messages that are responded to. This count is set to 0 when the UDPEcho server is enabled.

The time information of UDP Echo Plus packet is denoted as following:

- **Gsi :** It is the time the UDPEchoPlus client sends the request packet.

- **Gri** : It is the time the UDPEchoPlus client receives the response packet.
- **Rri** : It is the time the UDPEchoPlus server receives the request packet (TestRespRecvTimeStamp).
- **Rsi** : It is the time the UDPEchoPlus server sends the response packet (TestRespReplyTimeStamp).

IP performance parameters include packet loss, transfer delay and delay jitter. The test method of ith UDP Echo Plus request is computed as following:

4.1.1. TRANSFER DELAY

The UDP Echo Plus protocol can exclude the system issue of prior storing server. It causes more transaction time ($Rsi - Rri$) for round trip delay. UDP Echo Plus request packet (denoted as $RTDi$) is depend on the effective round trip delay. It can calculate as following:

$$RTDi = Gri - Gsi - (Rsi - Rri)$$

$RTDi$ specification using RT-PEPA in the following:

$$RTDi := Gri \overset{[D]}{L,T} Gsi \overset{[D]}{L,T} Rsi \overset{[D]}{L,T} Rri$$

Rsi and Rri can be replaced by *TestRespRecvTimeStamp* and *TestRespReplyTimeStamp* field of ith response packet.

4.1.2 TRANSFER DELAY ANALYSIS

The transfer delay test setup is set as following:

- No any downstream and upstream delay.
- Adding Upstream delay (10 ms).
- Adding Downstream delay (15 ms).
- Adding Downstream delay (15 ms) and Upstream delay (10 ms) simultaneously.

Table 1 show the test result. Test setup (a) is base of other test condition. The test error percentage is calculated by the difference with (a). For instance, the error percentage of **RoundTripDelayMean** is calculated as $[(15.639-5.619)-10]/10 = 0.19\%$.

The **RoundTripDelayMean** has absolute difference and percentage with the expected value. The **RoundTripDelayMaxth** would decrease this critical condition.

| Test Result | Round Delay Mean | Trip Min | Round Delay Max | Trip MaxTH (98%) |
|---------------|---------------------|-------------|--------------------|---------------------|
| (a) | 5.619 | 5.348 | 6.696 | 5.789 |
| (b) | 15.637 | 15.364 | 17.312 | 15.812 |
| (c) | 20.834 | 20.323 | 22.138 | 21.065 |
| (d) | 30.261 | 30.331 | 32.605 | 30.798 |
| (b)-(a) | 10.018 | 10.015 | 10.601 | 10.021 |
| (c)-(a) | 15.203 | 14.974 | 15.442 | 15.277 |
| (d)-(a) | 25.003 | 24.982 | 25.907 | 25.012 |
| Error% of (b) | 0.17% | 0.16% | 6.01% | 0.22% |
| Error% of (c) | 1.35% | -0.16% | 2.96% | 1.84% |
| Error% of (d) | 0.01% | -0.06% | 3.64% | 0.04% |

Table 1: Transfer Delay Test Result

4.2.1. DELAY JITTER

UDP Echo Plus packet format provides the capability to calculate one-way IPTV Delay Jitter

- Delay Jitter of the i th UDP Echo Plus transaction on send path is computed with i -th transaction as IPTV Delay Jitters(i):

$$Send\ time\ delta = Gsi - Gsi-1$$

Send time delta Specification using RT-PEPA in the following:

$$Send\ time\ delta := Gsi \stackrel{D,T}{L,T} Gsi-1$$

$$Receive\ time\ delta = Rri - Rri-1$$

Receive time delta Specification using RT-PEPA using RT-PEPA in the following:

$$Receive\ time\ delta := Rri \stackrel{D,T}{L,T} Rri-1$$

$$Delay\ Jitters(i) = | Send\ time\ delta - Receive\ time\ delta |$$

Delay Jitter Specification using RT-PEPA in the following:

$$Delay\ Jitters(i) := Send\ time\ delta \stackrel{D,T}{L,T} Receive\ time\ delta$$

- IPTV Delay Jitter of the i th UDP Echo Plus transaction on receive path is calculated with i -th transaction as $Delay\ Jitters(i)$:

$$Response\ time\ delta = Rsi - Rsi-1$$

Response time delta Specification using RT-PEPA in the following:

$$Response\ time\ delta := Rsi \stackrel{D,T}{L,T} Rsi-1$$

$$Receive\ time\ delta = Gri - Gri-1$$

Receive time delta Specification using RT-PEPA in the following:

$$Receive\ time\ delta := Gri \overset{L,T}{\Delta} Gri-1$$

$$Delay\ Jitters(i) = | Response\ time\ delta - Receive\ time\ delta |$$

Delay Jitters (i) Specification using RT-PEPA in the following :

$$Delay\ Jitters(i) := Response\ time\ delta \overset{L,T}{\Delta} Receive\ time\ delta$$

(Rsi , Rri) and (Rsi-1 , Rri-1) can be replace by the ith and i-1th TestRespRecvTimeStamp and TestRespReplyTimeStamp field of UDP Echo Plus response packet.

4.2.2 DELAY JITTER ANALYSIS

The delay jitter setup is set as following:

(a) No any downstream and upstream jitter.

(b) Adding Upstream jitter, the delay jitter setup is transfer delay is from 5 to 25 ms, decreasing 0.5 ms constantly until delay is 5 ms, and reshifting to 25 ms. The expected UplinkJitterMax and UplinkJitterMean are 20 ms and 0.975 ms individually

(c) Adding Downstream jitter, jitter setup is same as (b).

Table 2 shows the test result. Test result in test setup (a) shows the essential jitter movement, no any jitter is added.

The critical upstream jitter (UplinkJitterMax) is reaches to 20.434ms, and has the 2.16% deviation with the expected value (20 ms). The DownlinkJitterMaxth and UplinkJitterMaxth would smooth the critical situation.

| Test Result | Uplink Jitter | Uplink Jitter | Uplink Jitter | Downlink Jitter | Downlink Jitter | Downlink Jitter |
|-------------|---------------|---------------|---------------|-----------------|-----------------|-----------------|
| | Mean | Max | MaxTH(98%) | Mean | Max | MaxTH(98%) |
| (a) | 0.132 | 3.987 | 0.221 | 0.077 | 2.117 | 0.123 |
| (b) | 0.994 | 20.434 | 0.974 | 0.104 | 2.368 | 0.143 |
| (c) | 0.114 | 2.293 | 0.198 | 0.991 | 20.121 | 0.952 |

Table 2: Delay Jitter Test Result

4.3.1. PACKET LOSS

UDP Echo Plus protocol provides to compute one-way packet loss ratio (PLR) and round trip.

- Round trip PLR: The round trip packet loss count represents as PLRT , and the ratio represent as PLRRT , and calculated as following:

$$PLRT = TestGenSN - SuccessfulEchoCnt$$

PLRT Specification using RT-PEPA in the following:

$$PLRT := TestGenSN \overset{L,T}{\Delta} SuccessfulEchoCnt$$

$$PLRRT = PLRT / TestGenSN$$

PLRRT Specification using RT-PEPA in the following:

$$PLR_{RT} := PL_{RT} \frac{\text{TestGenSN}}{\text{TestRespSN}}$$

- Send path PLR: Sent packet loss is determined by subtracting the total requests received by the UDPEcho Plus server (**TestRespSN**) from the total request sent by the UDPEchoPlus client (**TestGenSN**). The send path packet loss count represents as PL_{SP} , and the ratio represents as PLR_{SP} , and computed as following:

$$PL_{SP} = \text{TestGenSN} - (\text{TestRespSN} + \text{TestRespReplyFailureCount})$$

PL_{SP} specification using RT-PEPA in the following :

$$PL_{SP} = \text{TestGenSN} \frac{\text{TestRespSN}}{\text{TestRespReplyFailureCount}}$$

$$PLR_{SP} = PL_{SP} / \text{TestGenSN}$$

PLR_{SP} specification using RT-PEPA in the following :

$$PLR_{SP} := PL_{SP} \frac{\text{TestGenSN}}{\text{TestRespSN}}$$

- Receive path PLR: Receive packet loss can calculate upon receiving a UDPEchoPlus response packet from the UDPEchoPlus server at some time Gri, The receive path packet loss represents as PL_{RP} , and the ratio represents as PLR_{RP} , and calculated as following:

$$PL_{RP} = PL_{RT} - PL_{SP}$$

PL_{RP} specification using RT-PEPA in the following:

$$PL_{RP} := PL_{RT} \frac{\text{TestRespSN}}{\text{TestGenSN}} - PL_{SP}$$

$$PLR_{RP} = PL_{RP} / \text{TestRespSN}$$

PLR_{RP} specification using RT-PEPA in the following:

$$PLR_{RP} = PL_{RP} \frac{\text{TestGenSN}}{\text{TestRespSN}}$$

5.3.2 PACKET LOSS ANALYSIS

The packet loss test setup is set as following:

- No packet loss occurs on both directions.
- 10% packet loss occurs on upstream.
- 10% packet loss occurs on downstream.
- 10% packet loss occurs on both directions concurrently.

Then the proposed QoS test method is executed and get the packet loss information for upstream ($UplinkPacketLoss$), and downstream ($DownlinkPacketLoss$) and packet loss of round trip ($RoundTripPacketLoss$). The test results of four test setups can expect.

5. CONCLUSIONS

In order to increase data transfer rate of mobile video for LTE system, the mobile carriers should improve Video-on-Demand (VoD) System. The main problem for Video-on-Demand (VoD) System is bandwidth limitation and storage. An efficient solution to solve this problem can use distributed content prior stores and prior store popular content. In this paper, we suggest the new architecture for LTE Network and Partial Prior Storage Servers. In this structure, we propose a partial prior storing strategy to store Video Content. The Prior Storage server is divided into two

level storages (Prior Storage 1 and Prior Storage 2) . The Video Content depend on popularity characteristics of the video content and the value of utility to determine prior storage 1 or prior storage. Prior Storage 1 (primary prior storage) is used to store a video content segment that is accessed for the first time and Prior Storage 2 (secondary prior storage) is used to store segments whose utility is already determined to be high (i.e., popular over a longer period of time). The distributed local video content servers allow operators to economically alleviate the inherent storage and network bandwidth limitation, proportionally distribute the subscriber load and service demand.

This project aims to improve the performance of LTE Network. In this project, we propose new architecture for servers. Then, we use RT-PEPA as a tool to simulate the behavior of LTE Network and analyse the performance of LTE Network. The Simulation Test results show that our proposed new architecture can improve the system performance and QoS parameters in terms of packet loss, packet delay and jitter.

REFERENCES

- [1] Jun He, Honghai Zhang, Baohua Zhao, and Sampath Rangarajan. 2013, A collaborative framework for in-network video caching in mobile networks. In Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference
- [2] Jun He, Xiaoming Zhao, and Baohua Zhao. 2012, A fast, simple and near optimal content placement scheme for a large-scale vod system. In Communication Systems (ICCS), 2012 IEEE International Conference on, pages 378–382.
- [3] Jun He, Xiaoming Zhao, Baohua Zhao, 2012, A Fast, Simple and Near-optimal Content Placement Scheme for a Large-scale VoD System, IEEE ICCS
- [4] Jun He, Xiaoming Zhao, Baohua Zhao, 2013, Joint Request Routing and Video Adaptation in Collaborative VoD Systems, IEEE Wireless Communications and Networking Conference (WCNC); NETWORKS
- [5] Hao Wang, David I. Laurenson and Jane Hillston, Evaluation of RSVP and Mobility-aware RSVP Using Performance Evaluation Process Algebra
- [6] Tony Tsang 2015, Video Contents Prior Storing Server for Optical Access Network, International Journal of Computer Networks & Communications (IJCNC) Vol.7, No.2.
- [7] Tony Tsang 2013, Performance Analysis for LTE Networks with Markov Decision Process, CYBER JOURNALS: MULTIDISCIPLINARY JOURNALS IN SCIENCE AND TECHNOLOGY, JOURNAL OF SELECTED AREAS IN TELECOMMUNICATIONS (JSAT), AUGUST EDITION, 2013, VOLUME 3, ISSUE 8.
- [8] Tony Tsang 2012, A Method for Performance Modelling and Evaluation of LDPC Decoder Architecture, IEEE

AUTHORS

Tony Tsang (MIEEE'2000) received the BEng degree in Electronics & Electrical Engineering with First Class Honours in U.K., in 1992. He received the Ph.D from the La Trobe University (Australia) in 2000. He was awarded the La Trobe University Post-graduation Scholarship in 1998. Prior to joining the Hong Kong Polytechnic University, Dr. Tsang earned several years of teaching and researching experience in the Department of Computer Science and Computer Engineering, La Trobe University. He works in Hong Kong Polytechnic University as Lecturer since 2001. He works in Hong Kong College of Technology in 2015. He has numerous publications in international journals and conferences and is a technical reviewer for several international journals and conferences. His research interests include mobile computing, networking, protocol engineering and formal methods. Dr. Tsang is a member of the IET and the IEEE.

