

NEWSMAN: Uploading Videos over Adaptive Middleboxes to News Servers In Weak Network Infrastructures

Rajiv Ratn Shah¹, Mohamed Hefeeda², Roger Zimmermann¹, Khaled Harras³,
Cheng-Hsin Hsu⁴, and Yi Yu⁵

¹ National University of Singapore, Singapore

{rajiv, rogerz}@comp.nus.edu.sg

² Qatar Computing Research Institute, Qatar

mhefeeda@qf.org.qa

³ CMU in Qatar, Qatar

yiyu@nii.ac.jp

⁴ National Tsinghua University, Taiwan

kharras@qatar.cmu.edu

⁵ National Institute of Informatics, Japan

chsu@cs.nthu.edu.tw

Abstract. An interesting recent trend, enabled by the ubiquitous availability of mobile devices, is that regular citizens report events which news providers then disseminate, *e.g.*, CNN iReport. Often such news are captured in places with very weak network infrastructure and it is imperative that a citizen journalist can quickly and reliably upload videos in the face of slow, unstable, and intermittent Internet access. We envision that some middleboxes are deployed to collect these videos over energy-efficient short-range wireless networks. Multiple videos may need to be prioritized, and then optimally transcoded and scheduled. In this study we introduce an adaptive middlebox design, called NEWSMAN, to support citizen journalists. NEWSMAN jointly considers two aspects under varying network conditions: (i) choosing the optimal transcoding parameters, and (ii) determining the uploading schedule for news videos. We design, implement, and evaluate an efficient scheduling algorithm to maximize a user-specified objective function. We conduct a series of experiments using trace-driven simulations, which confirm that our approach is practical and performs well. For instance, NEWSMAN outperforms the existing algorithms: (i) by 12 times in terms of system utility (*i.e.*, sum of utilities of all uploaded videos), and (ii) by 4 times in terms of the number of videos uploaded before their deadline.

Keywords: Video transcoding, adaptive transmission, news reporting, news scheduling, multimedia information systems, information systems applications

1 Introduction

Owing to technical advances in mobile devices and wireless communications, user-generated news videos have become popular since they can be easily captured using most modern smartphones and tablets in sufficiently high quality. Moreover, in the era of globalization, most news providers cover news from every part of the world, while

on many occasions, reporters send news materials to editing rooms over the Internet. Therefore, in addition to traditional news reporting, the concept of citizen journalism, which allows people to play active roles in the process of collecting news reports, is also gaining much popularity. For instance, CNN allows citizens to report news using modern smartphones and tablets through its CNN iReport service. This service has more than one million citizen journalist users [2], who report news from places where traditional news reporters may not have access. Every month, it garners an average of 15,000 news reports and its content nets 2.6 million views [1]. It is, however, quite challenging for reporters to timely upload news videos, especially from developing countries, where Internet access is slow or even intermittent. Hence, it is crucial to deploy *adaptive middleboxes*, which upload news videos respecting the varying network conditions. Such middleboxes will allow citizen reporters to quickly *drop* the news videos over energy-efficient short-range wireless networks, and continue their daily life.

Journalists can upload news videos to middleboxes or news providers, either by using cellular or WiFi networks if available. Since an energy-efficient short-range wireless network between mobile devices and middleboxes can be leveraged using optimized mobile applications, we focus on a scheduling algorithm tuned for varying network conditions which can adaptively schedule the uploads of videos. Middleboxes can be placed in cloud servers or strategic places in towns such as city centers, coffee shops, train and bus stations, *etc.*, so that when reporters frequent these places then the short-range wireless communication can be leveraged for uploading videos. One can envision that an efficient smartphone application can further improve such communication among different reporters based on collaborative models. Shops at these places may host such middleboxes incentivized by the following reasons: (i) advertisement companies can sponsor the cost of resources (*e.g.*, several companies already sponsor Internet connectivity at airports), (ii) news providers can sponsor resources since they will receive news on time with less investment, (iii) more customers may be attracted to visit these shops, and (iv) a collaborative model of information sharing based on crowdsourcing is gaining popularity. Moreover, middleboxes can be used to decide whether reporters can directly upload videos to news providers based on current network conditions.

In designing the adaptive middlebox, we consider two categories of news videos, first, *breaking news* and, second, *traditional news*. Usually, the breaking news videos have stricter deadlines than those of the traditional news videos. There is significant competition among news organizations to be the first to report breaking news. Hence, ubiquitous availability of mobile devices and the concept of citizen journalism help with fast reporting of news videos, using the mobile applications and the web sites of news providers. However, many times, the uploading of news videos is delayed due to reporters' slow Internet access and the big sizes of news videos. In pilot experiments among news reporters in early 2015, we noticed low throughput and non-trivial network interruptions in some of our test cases, as summarized in Table 1. Reporters tested uploading from a few locations in India, Pakistan, Argentina, and the USA, mostly through cellular networks. For example, when news reporters uploaded their videos over the Internet to an editing room in New York City for a leading news provider, they suffered from as many as 7 interrupts per upload. Without our proposed adaptive

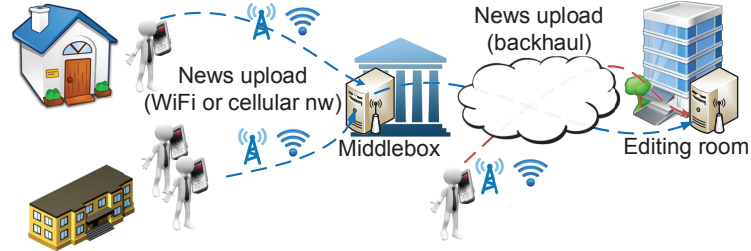


Fig. 1: Architecture of the proposed NEWSMAN system.

Table 1: Real world results of news uploading.

Location	India	Pakistan	Argentina	USA
Throughput	500 ~ 600 Kbps	300 ~ 500 Kbps	200 ~ 300 Kbps	20 ~ 23 Mbps
File Sizes	100 ~ 200 MB	50 ~ 100 MB	500 ~ 600 MB	100 ~ 200 MB
# of Interruptions	6	3	7	0

middleboxes, news reporters may be frustrated and eventually give up, because of long uploading times. This necessitates carefully designed adaptive middleboxes which run a scheduling algorithm to determine an uploading schedule for news videos considering factors such as optimal bitrates, videos deadlines, and network conditions.

In this study, we propose NEWSMAN, which maximizes the system utility by optimizing the number and quality of the videos uploaded before their deadlines from users to news editors under varying network conditions. We place middleboxes between reporters and news editors, to de-couple the local upload from the long-haul transmission to the editing room, in order to optimize both network *segments*, which have diverse characteristics. To optimize the system performance, we design an efficient scheduling algorithm in the middlebox to derive the uploading schedule and to transcode news videos (if required, to meet their deadlines) adaptively following a practical video quality model. The NEWSMAN scheduling process is described as follows: (i) reporters directly upload news videos to the news organizations if the Internet connectivity is good, otherwise (ii) reporters upload news videos to the middlebox, and (iii) the scheduler in the middlebox determines an uploading schedule and optimal bitrates for transcoding. Figure 1 presents the architecture of the NEWSMAN system.

The key contribution of this study is an efficient scheduling algorithm to upload news videos to a cloud server such that: (i) the system utility is maximized, (ii) the number of news videos uploaded before their deadlines is maximized, and (iii) news videos are delivered in the best possible video qualities under varying network conditions. We conducted extensive trace-driven simulations using real datasets of 130 online news videos. The results from the simulations show the merits of NEWSMAN as it outperforms the current algorithms: (i) by 1,200% in terms of system utility and (ii) by 400% in terms of the number of videos uploaded before their deadlines. Furthermore, NEWSMAN achieves low average delay of the uploaded news videos. The remaining parts of this paper are organized as follows. In Section 2, we review the related literature, and we describe the NEWSMAN system in Section 3. We present and solve the upload scheduling

problem in Section 4. The experiments and results are presented in Section 5. Finally, we conclude the paper with a summary in Section 6.

2 Related Work

The NEWSMAN scheduling process is described as follows, (i) reporters directly upload news videos to the news organizations if the Internet connectivity is good, otherwise (ii) reporters upload news videos to a middlebox, and (iii) the scheduler at the middlebox determines an uploading schedule and optimal bitrates for transcoding. In this section, we survey some recent related work.

In addition to traditional news reporting systems such as satellite news networks, the use of satellite news gathering (SNG) by local stations has also increased during recent years. However, SNG has not been adopted as widely as satellite news networks due to reasons such as: (i) the high setup and maintenance costs of SNG, and (ii) the non-portability of SNG equipment to many locations due to its big size [9, 14]. These constraints have popularized news reporting by regular citizens through services such as CNN iReport [1].

Unlike significant efforts that have focused on systems supporting downloading applications such as video streaming and file sharing [12, 17], little attention has been paid to systems that support uploading applications [4, 19]. Media uploading with hard deadlines require an optimal deadline scheduling algorithm [3, 5, 18]. Abba *et al.* [3] proposed a prioritized deadline based scheduling algorithm using a project management technique for an efficient job execution with deadline constraints of jobs. Chen *et al.* [5] proposed an online preemptive scheduling of jobs with deadlines arriving sporadically. The scheduler either accepts or declines a job immediately upon arrival based on a contract where the scheduler loses the profit of the job and pays a penalty if the accepted job is not finished within its deadline. The objective of the online scheduler is to maximize the overall profit, *i.e.*, the total profit of completed jobs before their deadlines is more than the penalty paid for the jobs that missed their deadlines. Online scheduling algorithms such as earliest deadline first (EDF) [13] are often used for applications with deadlines. Since we consider jobs with diverse deadlines, we leverage the EDF concept in our system to determine the uploading schedule that will maximize the system utility.

Recent years have seen significant progress in the area of rate-distortion (R–D) optimized image and video coding [6, 10]. In lossy compression, there is a tradeoff between the bitrate and the distortion. R–D models are functions that describe the relationship between the bitrate and expected level of distortion in the reconstructed video. In NEWSMAN, R–D models enable the optimization of the received video quality under different network conditions. To avoid unnecessary complexity of deriving R–D models of individual news videos, NEWSMAN categorizes news videos into a few classes using *temporal perceptual information* (TI) and *spatial perceptual information* (SI), which are the measures of temporal changes and spatial details, respectively [7, 16]. Due to limited storage space, less powerful CPU, and constrained battery capacity, earlier works [8, 11] suggested to perform transcoding at resourceful clouds (middleboxes in our case) instead of at mobile devices. In our work we follow this model.

3 System Overview

We refer to the uploading of a news video as a *job* in this study. NEWSMAN schedules jobs such that videos are uploaded before their deadlines in the highest possible qualities with optimally selected coding parameters for video transcoding.

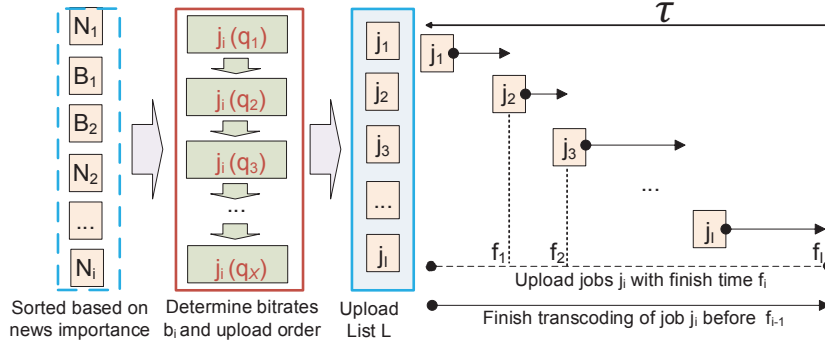


Fig. 2: Scheduler architecture in a middlebox.

NEWSMAN Scheduling Algorithm. Figure 2 shows the architecture of the scheduler. Reporters upload jobs to a middlebox. For every job arriving at the middlebox, the scheduler performs the following actions when the scheduling interval expires: (i) it computes the job’s importance, (ii) it sorts all jobs based on news importance, and (iii) it estimates the job’s uploading schedule and the optimal bitrate for transcoding. The scheduling algorithm is described in details in Section 4. As Figure 2 shows, we consider χ video qualities for a job j_i and select the optimal bitrate for transcoding of j_i to meet its deadline, under current network conditions.

R-D Model. Traditional digital video transmission and storage systems either fully upload a news video to a news editor or not at all. The key idea for transcoding videos with optimal bitrates is to compress videos for transmission to adaptively transfer video contents before their deadlines, under varying network conditions. More motion in adjacent frames indicates higher TI values and scenes with minimal spatial detail result in low SI. For instance, a scene from a football game contains a large amount of motion (*i.e.*, high TI) as well as spatial detail (*i.e.*, high SI). Since two different scenes with the same TI/SI values produce similar perceived quality [16], news videos can be classified in G categories. Therefore, news videos can be categorized into different categories such as sport videos, interviews, *etc.*, based on their TI/SI values. It is important to determine the category (or TI/SI values) of a news video, so that we can select appropriate R–D models for these categories. A scene with little motion and limited spatial detail (such as a head and shoulders shot of a newscaster) may be compressed to 384 kbits/sec and decompressed with relatively little distortion. Another scene (such as from a football game) which contains a large amount of motion as well as spatial detail will appear quite distorted at the same bit rate [16]. Therefore, it is important to consider different R–D models for all categories. Empirical piecewise linear R–D models can be constructed for individual TI/SI pairs. We encode online news videos with diverse content complexities and empirically analyze their R–D characteristics. We consider four categories in our experiments corresponding to *high TI/high SI*, *high TI/low SI*, *low*

TI/high SI, and *low TI/low SI*. We adaptively determine the suitable coding bitrates for an editor-specified video quality for videos, using these piecewise linear R–D models.

4 Scheduling Problem and Solution

Formulation. Let there be B breaking news and N traditional news, with given arrival times A , deadlines D , and metadata M which consist of users' reputations and video metadata such as bitrates and fps (frames per second). For a job j_i , let $\xi(j_i)$ and $\lambda(j_i)$ be scores for video length and news-location, respectively. Let $\gamma(r)$ be the user reputation score for a reporter r . Let σ be the editor-specified minimum required video quality (say in PSNR, peak signal-to-noise ratio). Let \bar{p}_i , p_i , \bar{b}_i and b_i be the original video quality, transcoded video quality, original bitrate, and transcoded bitrate for job j_i , respectively. Let $\omega(t_c)$ be the available disk size at some time t_c . Let \bar{s}_i and s_i be the original and transcoded file sizes, respectively. Let $\eta(s_i)$ be the time required to transcode the job j_i with file size s_i and $\beta(t_1, t_2)$ the average throughput between time interval t_1 and t_2 . Let $\delta(j_i)$ be the video length (in seconds) for j_i . Let τ be the time interval of running the scheduler in a middlebox.

The news importance u of a job j_i is defined as $u(j_i) = \mu(j_i) \cdot (w_1 \xi(j_i) + w_2 \lambda(j_i) + w_3 \gamma(r))$, where the multiplier $\mu(j_i)$ is a weight for boosting or ignoring the importance of any particular news type or category. E.g., in our experiments the value of $\mu(j_i)$ is 1 if job j_i is traditional news and 2 if job j_i is breaking news. By considering news categories such as sports a news provider can boost videos during a sports events such as the FIFA world cup. Moreover, the news decay function v is defined as:

$$v(f_i) = \begin{cases} 1, & \text{if } f_i \leq d_i \\ e^{-\alpha(f_i - d_i)}, & \text{otherwise, where } f_i \text{ and } d_i \text{ are the finish time and deadline of} \\ & \text{job } j_i, \text{ respectively. } \alpha \text{ is an exponential decay constant.} \end{cases}$$

The utility score of a news video j_i depends on the following factors: (i) the importance of j_i , (ii) how quickly the importance of j_i decays, and (iii) the delivered video quality of j_i . Thus, we define the news utility ρ for job j_i as $\rho(j_i) = u(j_i) v(f_i) p_i$. With the above notations and functions, we state the problem formulation as:

$$\max \sum_{i=1}^{B+N} \rho(j_i) \quad (1a)$$

$$s.t. \quad \sigma \leq p_i \leq \bar{p}_i \quad \forall 1 \leq i \leq B+N \quad (1b)$$

$$(f_i - f_{i-1}) \beta(f_{i-1}, f_i) \geq b_i \delta(j_i) \quad (1c)$$

$$\sum_{\forall j_k \in K} \eta(s_k) < f_i, \text{ where } K = \{j_k \mid j_k \text{ is scheduled before } j_i\} \quad (1d)$$

$$\sum_{i=1}^{B+N} s_i \leq \omega(t_c) - \sum_{i=1}^{B+N} \bar{s}_i \quad (1e)$$

$$f_i \leq f_k, \quad \forall 1 \leq i \leq k \leq B+N \quad (1f)$$

$$0 \leq f_i, \quad \forall 1 \leq i \leq B+N \quad (1g)$$

$$0 < b_i \leq \bar{b}_i, \quad \forall 1 \leq i \leq B+N \quad (1h)$$

$$j_i \in \{1, \dots, B, B+1, B+2, \dots, B+N\} \quad (1i)$$

The objective function in Eq. (1a) maximizes the sum of news utility (*i.e.*, the product of importance, decay value and video quality) for all jobs. Eq. (1b) makes sure that the video quality of the transcoded video is at-least the minimum video quality σ . Eq. (1c) ensures bandwidth constraints for NEWSMAN. Eq. (1d) enforces that the transcoding of a video completes before its uploading starts and Eq. (1e) ensures disk constraints of a middlebox. Eq. (1f) ensures that the scheduler uploads jobs in the order scheduled by NEWSMAN. Eqs. (1g) and (1h) define the ranges of the decision variables. Finally, Eq. (1i) indicates that all jobs are either breaking news or traditional news.

Lemma. Let $j_{i=1}^n$ be a set of n jobs in a middlebox at time t_c , and $d_{i=1}^n$ their respective deadlines for uploading. The scheduler is executed when either the scheduling interval τ expires or when all jobs in the middlebox have been uploaded before τ expires. Thus, the average throughput $\beta(t_c, t_c + \tau)$ (or β in short) during the scheduling interval is distributed among several jobs selected for parallel uploading⁶, and as a consequence, the sequential upload of jobs has higher utility than parallel uploading.

Proof Sketch. Let k jobs $j_{i=1}^k$ with transcoded sizes $s_{i=1}^k$ be selected in parallel uploading. Let k_t of them ($j_{i=1}^{k_t}$) require transcoding. Thus, it takes some time for their transcoding (*i.e.*, $\eta_p(s_i)_{i=1}^{k_t} \geq 0$) before the actual uploading starts. Hence, uploading throughput is wasted during the transcoding of these jobs in parallel uploading. During sequential uploading, NEWSMAN ensures that transcoding of a job is finished (if required) before the uploading of the job is started. Thus, it results in a net transcoding time of zero (*i.e.*, $\eta_s(s_i)_{i=1}^{k_t} = 0$) in sequential uploading, and it fully utilizes the uploading throughput β . Let t_u be the time (excluding transcoding time) to upload jobs $j_{i=1}^n$. Thus, t_u is equal for both sequential and parallel uploading since the same uploading throughput is divided among parallel jobs. Let t_p (*i.e.*, $t_u + \eta_p$) and t_s (*i.e.*, $t_u + \eta_s$) be the uploading time for all jobs ($j_{i=1}^n$) when the jobs are uploaded in parallel or sequential manner, respectively. Hence, the actual time required to upload in a parallel manner (*i.e.*, t_p) is greater than the time required to upload in a sequential manner (*i.e.*, t_s). Moreover, the uploading of important jobs is delayed in parallel uploading since throughput is divided among several other selected jobs (β/k for each job). Therefore, the sequential uploading of jobs is better than the parallel uploading.

Upload Scheduling Algorithm. We design an efficient scheduling algorithm to solve the above formulation. Alg. 1 shows the main procedure of scheduling a list of jobs at a middlebox. If it is not possible to upload any job within its deadline, NEWSMAN uploads the transcoded news videos to meet the deadline. Alg. 2 shows the procedure of calculating the encoding parameters for transcoding under current network conditions and σ . Alg. 2 is invoked on line 18 of Alg. 1 whenever necessary. The NEWSMAN scheduler considers χ possible video qualities (hence, smaller video size and shorter upload time are possible) for a job. NEWSMAN considers σ as a threshold and divides a region between σ (minimum required video quality) and \bar{p}_i (original video quality) among χ discrete qualities (say, $q_{i=1}^\chi$, with $q_1 = \sigma$ and $q_\chi = \bar{p}_i$). The scheduler keeps checking lower, but acceptable, video qualities starting with the least important job first, to accommodate j in L such that: (i) the total estimated system utility increases

⁶ Some videos may require transcoding first before uploading to meet deadlines in NEWSMAN.

Algorithm 1 A procedure for scheduling news videos

```

1: procedure SCHEDULEANDUPLOADNEWSREPORTS
2:   INPUT:  $B + N$  jobs with given arrival times  $A$  and deadlines  $D$ 
3:   OUTPUT: A uploading schedule for jobs at a middlebox
4:   Initialization:  $t = t_c, \mathcal{U} = 0$   $\triangleright t_c$  is current time and  $\mathcal{U}$  is total utility.
5:    $Q = \text{getAllJobs}(t_c)$   $\triangleright Q$  is the list of all jobs arrived till time  $t_c$  at the middlebox.
6:    $\text{compNewsImportance}(Q)$   $\triangleright$  Compute news importance for all jobs in  $Q$ .
7:    $\text{compEstmFinishTime}(Q, \chi)$   $\triangleright$  Compute estimated finish time for  $\chi$  video qualities.
8:    $\text{sortJobs}(Q)$   $\triangleright$  Sort all jobs in  $Q$  based on their news importance scores.
9:   for each job  $j_i$  in  $Q$  do  $\triangleright$  Read a job with the highest importance score first.
10:     $t_d = \text{getDeadline}(j_i, A)$   $\triangleright$  Get the deadline (maximum allowed finish time) for  $j_i$ .
11:     $t_h = \text{getEstmFinishTime}(j_i, t)$   $\triangleright$  Estimated time to upload  $j_i$  in original quality.
12:    if ( $t_h \leq t_d$ ) then  $\triangleright$  Indicates  $j_i$  can be uploaded in original video before deadline.
13:       $\rho(j_i) = u(j_i) v(t_h) p_i$   $\triangleright$  Utility based on its importance, decay and quality.
14:       $\mathcal{U} += \rho(j_i)$   $\triangleright$  Total utility for jobs scheduled so far.
15:       $\text{addJobToUploadList}(j_i, L)$   $\triangleright$  Added  $j_i$  to  $L$  in original video quality.
16:       $t += t_h$   $\triangleright$  Update total scheduled time so that more jobs can be added to  $L$ .
17:    else  $\triangleright$  Try to add  $j_i$  to  $L$  in lower video quality, if possible.
18:       $\phi = \text{addJobInLowerQuality}(j_i, t_d, t, L, \mathcal{U})$   $\triangleright$  See Alg. 2.
19:    end if  $\triangleright \phi$  is TRUE if  $j_i$  added to  $L$  after lowering qualities of some videos.
20:    if ( $\phi == \text{FALSE}$ ) then  $\triangleright \phi$  is FALSE if  $j_i$  could not be added to  $L$ .
21:       $\text{addJobToScheduleLater}(j_i, Q')$   $\triangleright$  Jobs in  $Q'$  could not be scheduled.
22:    end if
23:  end for  $\triangleright$  Output the uploading order and  $b_i$  (decision variables in formulation) of all  $j_i$ .
24:   $\text{transcodeAndUpload}(L)$   $\triangleright$  Set  $f_i$  (decision variable in formulation) for  $j_i$  after upload.
25: end procedure

```

after adding j , and (ii) all jobs in L still meet their deadlines (maybe with lower video qualities), if they are estimated to meet deadlines earlier. However, if the scheduler is not able to add j in the uploading list, then this job is added to a missed-deadline list whose deadline can be modified later by news-editors based on news importance. Once the scheduling of all jobs is done, NEWSMAN starts uploading news videos from the middlebox to the editing room and transcodes (in parallel with uploading) the rest of the news videos (if required) in the uploading list L .

Alg. 2 is invoked when it is not possible to add a job with the original video quality to L . This procedure keeps checking jobs at lower video qualities until all jobs in the list are added to L with estimated uploading times within their deadlines. The *isJobAccommodatedWithinDeadline()* method on line 13 of Alg. 2 ensures that: (i) the selected video quality q_k is lower than the current video quality q_c (*i.e.*, $q_k \leq q_c$) since some jobs are already set to lower video qualities in earlier steps, (ii) the utility value is increased after adding the job (*i.e.*, $\bar{\mathcal{U}} \geq \mathcal{U}$), (iii) all jobs in L is completed (estimated) within their deadlines, and (iv) a job with higher importance comes first in L .

5 Simulations and Results

Real-Life Datasets. We collected 130 online news video sequences from Al Jazeera, CNN, and BBC YouTube channels during mid-February 2015. The shortest and longest

Algorithm 2 A procedure for determining transcoding parameters for jobs in L , if j is added to the uploading list

```

1: procedure ADDJOBINLOWERQUALITY
2:   Initialization:  $\bar{t} = t, \bar{U} = U$   $\triangleright$  Initialize updated utility and scheduled time after adding  $j$ .
3:   addJobToUploadList( $j, L$ )  $\triangleright$  Add  $j$  to  $L$  in original quality and reduce quality later.
4:    $n = \text{getNumberOfJobs}(L)$   $\triangleright$  Check all existing jobs in  $L$  to determine scheduling list.
5:   for ( $m = n; m \geq 1; m--$ ) do
6:     for each  $q_k$  of  $j_m$  do  $\triangleright q_k$  is  $k^{\text{th}}$  video quality of  $j_m$ .
7:        $t_k = \text{getEstmFinishTimeLowQty}(j_m, t, q_k)$   $\triangleright$  Time to upload  $j_m$  in  $k^{\text{th}}$  quality.
8:        $t_e = \text{getEstmFinishTimeLowQty}(j_m, t, q_c)$   $\triangleright$  Time to upload  $j_m$  in current qty.
9:        $\rho_k = u(j_m) v(t_k) q_k$   $\triangleright$  Compute the utility of  $j_m$  for  $k^{\text{th}}$  quality.
10:       $\rho_c = u(j_m) v(t_e) q_c$   $\triangleright$  Compute the utility of  $j$  for current quality.
11:       $\bar{t} = \text{getUpdatedScheduledTime}(j_m, t_k, t_e)$   $\triangleright$  Get the updated scheduled time.
12:       $\bar{U} = \text{getUpdatedUtility}(j_m, \rho_c, \rho_k, U, \bar{U})$   $\triangleright$  Updated total utility value.
13:       $\phi = \text{isJobAccommodatedWithinDeadline}(m, t_d, \bar{t}, t, L, U, \bar{U})$   $\triangleright$  All jobs in  $L$ 
      (including  $j$ ) can be scheduled within deadline after lower qualities of jobs (from  $j_m$  to  $j_n$ ).
14:      if ( $\phi == \text{TRUE}$ ) then  $\triangleright$  Set the video quality of  $j_m$  to  $q_k$ .
15:        updateTranscodingParam( $m, L, q_k$ )  $\triangleright$  Set quality of jobs ( $j_{m+1}$  to  $j_n$ ) to  $q_k$ .
16:         $U = \bar{U}$   $\triangleright$  Update total utility value after changing qualities of some jobs.
17:         $t = \bar{t}$   $\triangleright$  Update the total uploading time after scheduling all jobs in  $L$ .
18:        return TRUE
19:      end if
20:    end for
21:  end for
22:  removeJobFromUploadList( $j, L$ )  $\triangleright j$  is removed from  $L$  since it could not be scheduled.
23:  return FALSE  $\triangleright j$  could not be added to  $L$  despite lowering qualities of jobs.
24: end procedure

```

Table 2: Statistics of network traces.

Location	Dates	Avg. Throughput
Delhi	2015-03-12 to 2015-03-14	409 Kbps
Hyderabad	2015-03-14 to 2015-03-18	297 Kbps
Nanjing	2015-03-23 to 2015-03-27	1138 Kbps

duration of videos are 0.33 and 26 minutes, and the smallest and biggest news video sizes are 4 and 340 MB, respectively. We also collected network traces from different PCs across the globe, such as (Delhi and Hyderabad) India, and (Nanjing) China, which emulate middleboxes in our system. More specifically, we use IPERF [15] to collect throughput from the PCs to an Amazon EC2 server in Singapore (see Table 2). The news and network datasets are used to drive our simulator.

Simulator Implementation and Scenarios. We implemented a trace-driven simulator for NEWSMAN using Java. Our focus is on the proposed scheduling algorithm under varying network conditions. The scheduler runs once every scheduling interval τ (say, 5 mins) in our simulator. The scheduler reads randomly generated new jobs following the Poisson process. We consider 0.1, 0.5, 1, 5 and 10 per min as mean job arrival rate and randomly mark a job as breaking news or traditional news in our experiments. In

the computation of news importance for videos, we randomly generate a real number in $[0,1]$ for user reputations and location importance in simulations. We set deadlines for news videos randomly in the following time intervals: (i) $[1, 2]$ hours for breaking news, and (ii) $[2, 3]$ hours for traditional news. We implemented two baseline algorithms: (i) earlier deadline first (EDF), and (ii) first in first out (FIFO) scheduling algorithms. For fair comparisons, we run the simulations for 24 hours and repeat each simulation scenario 20 times. If not otherwise specified, we use the first-day network trace to drive the simulator. We use the same set of jobs (with the same arrival times, deadlines, news types, user reputations, location importance, *etc.*) for three algorithms, in a simulation iteration. We report the average performance with 95% confidence intervals whenever applicable.

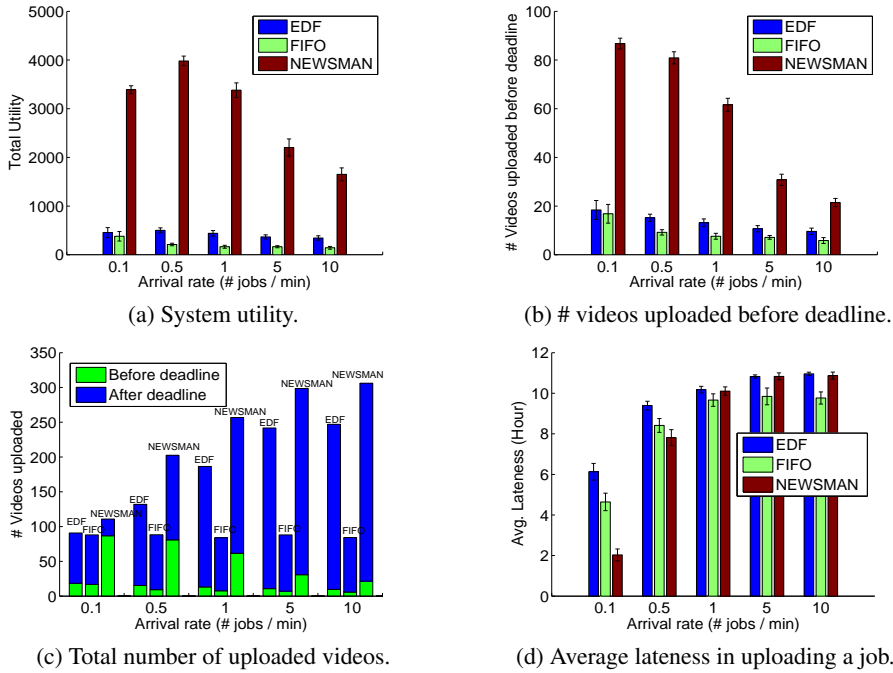


Fig. 3: Results after running the simulator for 24 hours using network traces from Delhi.

Results. *NEWSMAN delivers the most news videos in time, and achieves the highest system utility.* Figures 3a, 4a and 4c show that NEWSMAN performs up to 1200% better than baseline algorithms in terms of system utility. Figures 3b and 3c show that our system outperforms baselines (i) by up to 400% in terms of number of videos uploaded before their deadlines, and (ii) by up to 150% in terms of total number of uploaded videos. That is, NEWSMAN significantly outperforms the baselines either when news editors set hard deadlines (4X improvement) or soft deadlines (1.5X improvement).

NEWSMAN achieves low average lateness. Despite delivering the most news videos in time, and achieving the highest system utility for Delhi (see Figures 3 and 4), NEWSMAN achieves fairly low average lateness (see Figures 3d, 4b and 4d).

NEWSMAN performs well under all network infrastructures. Figure 4a shows that NEWSMAN outperforms baselines under all network conditions such as low average throughput in India, and higher average throughput in China (see Table 2).

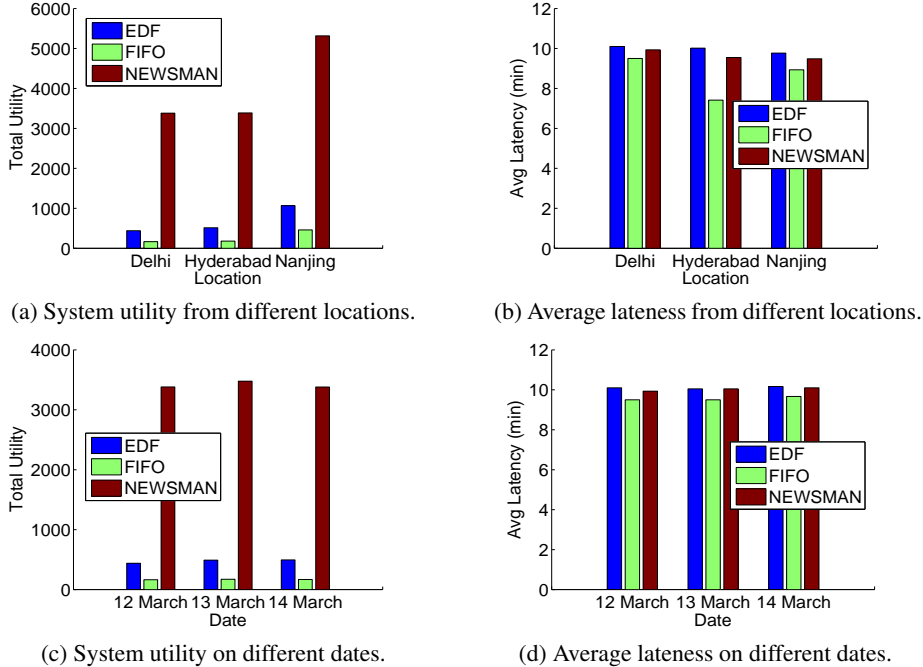


Fig. 4: Results after running the simulator for 24 hours using network traces from different locations (a–b) on different dates (c–d).

6 Conclusions

We present an innovative design for efficient uploading of news videos with deadlines under weak network infrastructures. In our proposed news reporting system called NEWSMAN, we use middleboxes with a novel scheduling and transcoding selection algorithm for uploading news videos under varying network conditions. The system intelligently schedules news videos based on their characteristics and underlying network conditions such that: (i) it maximizes the system utility, (ii) it uploads news videos in the best possible qualities, and (iii) it achieves low average lateness of the uploaded videos. We formulated this scheduling problem into a mathematical optimization problem. Furthermore, we developed a trace-driven simulator to conduct a series of extensive experiments using real datasets and network traces collected between a Singapore EC2 server and different PCs in Asia. The simulation results indicate that our proposed scheduling algorithm improves system performance. We are currently deploying NEWSMAN in developing countries to demonstrate its practicality and efficiency in practice.

References

1. iReport at 5: Nearly 900,000 contributors worldwide. <http://www.niemanlab.org/2011/08/ireport-at-5-nearly-900000-contributors-worldwide/>, July 2012. Online: Last Accessed March, 2015.

2. Meet the million: 999,999 iReporters + you! <http://ireport.cnn.com/blogs/ireport-blog/2012/01/23/meet-the-million-999999-ireporters-you>, July 2012. Online: Last Accessed March, 2015.
3. H. A. Abba, S. N. M. Shah, N. B. Zakaria, and A. J. Pal. Deadline based performance evaluation of job scheduling algorithms. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2012 International Conference on, pages 106–110. IEEE, 2012.
4. S. Bhattacharjee, W. C. Cheng, C.-F. Chou, L. Golubchik, and S. Khuller. Bistro: a framework for building scalable wide-area upload applications. *ACM SIGMETRICS Performance Evaluation Review*, 28(2):29–35, 2000.
5. S. Chen, L. Tong, and T. He. Optimal deadline scheduling with commitment. In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 111–118. IEEE, 2011.
6. M. Hefeeda and C.-H. Hsu. On burst transmission scheduling in mobile tv broadcast networks. *IEEE/ACM Transactions on Networking (TON)*, 18(2):610–623, 2010.
7. P. ITU-T RECOMMENDATION. Subjective video quality assessment methods for multimedia applications. 1999.
8. F. Jokhio, A. Ashraf, S. Lafond, I. Porres, and J. Lilius. Prediction-based dynamic resource allocation for video transcoding in cloud computing. In *International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 254–261. IEEE, 2013.
9. S. Lacy, T. Atwater, X. Qin, and A. Powers. Cost and competition in the adoption of satellite news gathering technology. *Journal of Media Economics*, 1(1):51–59, 1988.
10. P. Lambert, W. De Neve, P. De Neve, I. Moerman, P. Demeester, and R. Van de Walle. Rate-distortion performance of H. 264/AVC compared to state-of-the-art video codecs. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(1):134–140, 2006.
11. Z. Li, Y. Huang, G. Liu, F. Wang, Z.-L. Zhang, and Y. Dai. Cloud transcoder: Bridging the format and resolution gap between internet videos and mobile devices. In *International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 33–38. ACM, 2012.
12. C. Liang, Y. Guo, and Y. Liu. Is random scheduling sufficient in P2P video streaming? In *International Conference on Distributed Computing Systems*, pages 53–60. IEEE, 2008.
13. C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.
14. S. Livingston and D. A. V. BELLE. The effects of satellite technology on newsgathering from remote locations. *Political Communication*, 22(1):45–62, 2005.
15. A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf: The tcp/udp bandwidth measurement tool. <http://dast.nlanr.net/Projects/Iperf/>, 2005.
16. A. A. Webster, C. T. Jones, M. H. Pinson, S. D. Voran, and S. Wolf. Objective video quality assessment system based on human perception. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 15–26. International Society for Optics and Photonics, 1993.
17. D. Xie, B. Qian, Y. Peng, and T. Chen. A model of job scheduling with deadline for video-on-demand system. In *International Conference on Web Information Systems and Mining (WISM)*, pages 661–668. IEEE, 2009.
18. M. Zhang, J. Wong, W. Tavanapong, J. Oh, and P. de Groen. Media uploading systems with hard deadlines. In *International Conference on Internet and Multimedia Systems and Applications (IASTED)*, pages 305–310. Citeseer, 2004.
19. M. Zhang, J. Wong, W. Tavanapong, J. Oh, and P. de Groen. Deadline-constrained media uploading systems. *Multimedia Tools and Applications*, 38(1):51–74, 2008.