Research Article

Improving Performance in Dense Wireless Spaces by Controlling Bulk Traffic

Marat Zhanikeev

School of Management, Tokyo University of Science, Fujimi 1-11-2, Chiyoda-ku, Tokyo 102-0071, Japan

Correspondence should be addressed to Marat Zhanikeev; maratishe@gmail.com

Received 7 October 2016; Accepted 13 December 2016; Published 9 January 2017

Academic Editor: Changqiao Xu

Copyright © 2017 Marat Zhanikeev. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The growing number of wireless devices nowadays often results in congestion of wireless channels. In research, this topic is referred to as networking in dense wireless spaces. The literature on the topic shows that the biggest problem is the high number of concurrent sessions to a wireless access point. The obvious solution is to reduce the number of concurrent sessions. This paper proposes a simple method called Bulk-n-Pick which minimizes the number of prolonged concurrent sessions by separating bulk from sync traffic. Aiming at educational applications, under the proposed design, web applications would distribute the main bulk of content once at the beginning of a class and then rely on small messages for real time sync traffic during the class. For realistic performance analysis, this paper first performs real-life experiments with various counts of wireless devices, bulk sizes, and levels of sync intensity. Based on the experiments, this paper shows that the proposed Bulk-n-Pick method outperforms the traditional design even when only two concurrent bulk sessions are allowed. The experiment shows that up to 10 concurrent bulk sessions are feasible in practice. Based on these results, a method for online performance optimization is proposed and validated in a trace-based emulation.

1. Introduction

Wireless standards have recently developed from 802.11g [1] to 802.11n [2] which offers higher throughput and can use multiple antennas (MIMO), making it possible to achieve rates up to 300 Mbps. The development continues with 802.11ac standard with further improvements gradually entering practice. More details on the universe of wireless standards can be found in an excellent survey in [3]. Development of future 802.11 protocols, specifically the 802.11ax standard, can be found in [4] and is close in spirit to this paper in that the new protocol places the problem of wireless interference in the center.

Dense wireless space is a separate topic in recent literature [5]. It is part of the overall guidelines for WLAN deployment [6] in terms of both high density of client devices and dense packing of multiple Access Points (APs). For example, the study in [7] measured interferences in spaces with up to 50 APs and shows how interference depends specifically on *bulk size*, defined in this paper as simply the volume of transmitted information but also hinting at its relative size, among several other parameters (distance from AP, etc.). Several

recent research papers come with measurement results that are similar to the ones presented in this paper, specifically in that they support the notion of *abrupt deterioration* in performance when interference exceeds a given threshold [4, 8].

Wireless interference is also taken into account by research on *wireless beacons* [9]. The main advantage of beacons is that they can be used without association between clients and APs, which adds more flexibility in multi-AP spaces. However, when frame rate of beacons is increased and a driver/application hack is used to allow to stream data continuously over multiple beacon frames, interference also becomes a major problem.

This paper focuses on the case of *high density in client devices*. Specifically, an educational class with one AP and up to 20 client devices is used as the main application scenario in this paper. In Japan, there are guidelines that cover LAN deployment in schools but are silent about classes based on WLANs [10]. The problem in WLAN-based classes [11] is interference that causes *reliability problems* in web applications. A study in [12] discusses this problem based on real-life measurements and shows that web applications start having

reliability problems at 10+ and become difficult to manage at 20+ devices, under a level of traffic intensity which is common/standard in educational classes. Poor reliability here refers to HTTP requests which would return empty results due to failed or interrupted wireless sessions. The advice in [12] is to resolve the issue by *designing* web applications accordingly, that is, resending requests on failure and so forth. However, such a *resilience* feature has a side effect, where multiple resends translate into longer completion times for downloads, on average. Moreover, congestion in such cases can be further degraded when multiple clients are trying to recover interrupted connections at the same time. By comparison, the methods proposed in this paper resolve this congestion problem completely.

The core proposal in this paper is the *Bulk-n-Pick* method for multiparty (focus in this paper is on the one-to-many case) data exchange in dense wireless spaces. The name of the method can be read as *bulk-and-pick* and *bulk-then-pick*, both correctly identifying the important capabilities offered by the method. The core idea is to minimize wireless congestion by *scheduling bulk transfers* (or, in a broader sense, controlling them), which is covered by the *bulk* part of the method. The *pick* part refers to the *real time sync traffic* which references portions of the main (predownloaded) bulk. The above reliability problem is mostly removed because the *pick* traffic carries only small sync messages, thus reducing the probability of a high number of overlapping wireless sessions. Provided that the bulk is downloaded in the background (not making users wait), such a design supports a high level of interactivity in web applications running over the local wireless network.

The proposed design makes good practical sense in educational classes. Students can take turns downloading the bulks at the beginning of the class (or separate sessions/portions of class) and then enjoy reliable syncs during the class. Moreover, the optimization problem formulated in this paper generalizes this design to situations in which bulk can be downloaded freely at any time, while server side automatically reacts to and resolves wireless congestion as it occurs.

The specific contributions in this paper are as follows. A new experiment was conducted to fill the gaps found in the earlier dataset in [12]; with the new dataset, it is now possible to model a dense wireless space for both the traditional and the proposed methods. The practical metric in analysis is *completion time* which directly indicates how much time it requires for the entire class (assuming each student has its own wireless device) to complete the download of the main bulk. Results show that the class under the Bulk-n-Pick method can reduce the completion time of 100 Mbyte downloads (for all clients) down to 15 minutes versus over an hour for conventional classes.

Further analysis in this paper introduces the *density* metric which describes the frequency of successful downloads at a given time. Together with the *rate*, a metric describing the true throughput at application level, the optimization problem proposed in this paper can resolve congestion in wireless spaces in real time and with a high degree of flexibility.

This paper has the following structure. Section 2 discusses the research literature which, in addition to the above overview of the *dense wireless* topic itself, covers several adjacent areas of research. Section 3 introduces the core proposal referred to as the *Bulk-n-Pick* method, and Section 4 describes the dataset collected from real measurement experiments conducted in a mock wireless class; the dataset is used as the basis for trace-based analysis further in the paper. Section 5 presents simple analysis derived directly from the dataset. Section 6 formulates the optimization problem of the *decision rules* type, which makes it possible to run a wireless class in the online version of the Bulk-n-Pick mode, that is, when bulk traffic can be exchanged freely, without using a predefined schedule. Section 7 describes the setup for the emulation analysis of the proposed optimization method, results of which are discussed in Section 8. The paper is concluded in Section 9.

2. Related Work

The literature directly discussing the topic of *dense wireless spaces* was reviewed in Section 1. This section discusses the literature which does not deal with this topic directly but is either closely related to it or can be used as part of the method proposed in this paper. Earlier stage of this proposal can be found at [13]; that study conducted early tests of client-server coordination in wireless classrooms which provided further motivation and finally led to this proposal. Note that [13] only conducted preliminary and rather simple practical tests, while this paper proposes a generic optimization framework and analyzes performance over a wide variety of conditions.

Arguably, the closest related topic to dense wireless spaces is that of *wireless beacons*. Beacons refer to a technology that relies on the use of the beacon frame defined for all 802.11* standards. Beacon frames can normally hold only 256 bytes of information, but there are methods and even drivers implemented in hardware/software that can stream larger bulks via multiple consecutive beacon frames; this is referred to as beacon stuffing [9]. Beacons offer the obvious advantage: they work in the *broadcast* mode, while traditional dense wireless spaces are considered as unicast environments. The *broadcast* feature is also often referred to as *nonassociative* use of WiFi Access Points (AP)s; this is because the client device can listen to one or multiple beacons (APs) without having previously established a connection with each beacon. This is also an obvious convenience, since the default use of client devices today does not allow for multiple parallel connections to multiple APs. This paper will revisit the subject of beacons several times in this paper but, otherwise, places the subject out of scope. While this technology can offer a drastic decrease in wireless congestion, it would be difficult to implement it in practice at the current level of availability of custom-made devices. Future publications will repeatedly revisit the subject, following advances in hardware and software.

Unexpectedly, roadside infra, described by the 802.11p standard, has a strong relation to dense wireless spaces. There is current research that has already incorporated this relatively recent standard as part of the future vehicle-to-roadside

as well as vehicle-to-vehicle communications [14, 15]. To deal with the congestion at the roadside, the new methods adopt the same general approach that is found in *cognitive* and *opportunistic* wireless networks [16]. Since this is the part that is related to the topic of dense wireless spaces, we can ignore the vehicle-specific parts of the technology and focus on its cognitive/opportunistic parts. They are revisited further in this section.

4G+ wireless technologies are also closely related to the topic of dense wireless spaces. A good survey of 4G+ technologies can be found in [17]. 4G+ is mostly about the LTE-A suite of technologies. It is truly a suite of technologies that includes Coordinated Multipoint (CoMP), Multiple-Input-Multiple-Output (MIMO), Self-Organized Networks (SON), and others, each defined for a separate application in dense local wireless networks. Note that while 4G+ technology itself is still considered the wide-area access technology, via the concept of the *microcell* (eNodeB, femtocell, etc.), the *local-access* part of the technology also becomes important. The problem of (local) interference is considered to be a major issue in such networks [18], opening new venues for cognitive and opportunistic networking as part of 4G+ [16].

Here, it is important to understand the underlying nature of local connectivity in 4G+ networks. Device-to-Device (D2D) and Machine-to-Machine (M2M) technologies represent the peer-to-peer part of 4G+. However, while the true P2P assumes that the two devices are in direct communication with each other, the D2D/M2M technologies in 4G+ assume that the devices communicate via the microcells which are part of the larger 4G+ infrastructure. While it is not found in current literature, support for the true P2P would fully enable mobile clouds based on 4G+ suite of technologies. Given the CoMP and SON research, some steps are already being made in this direction.

Cognitive and opportunistic components are a major part of 4G+ and specifically the 5G technologies discussed in literature. Since microcells and traditional base stations play independent roles in establishing and maintaining wireless end-to-end paths, energy efficiency and, for terminals and spectrum, efficiency for the entire system are important factors. Discussions in [19] provide good background on energy/spectrum efficiency in the larger context of cognitive wireless networking. The term *spectral efficiency* in 4G+ research is the direct counterpart of the term *wireless congestion control* in this paper.

There is an obvious relation between the proposal in this paper and the cognitive/opportunistic approach. The latter achieves higher efficiency of wireless use by sensing and opportunistically putting to use openings in both time and spectrum. In this paper, we know in advance that dense wireless spaces (like educational classes as the main example in this paper) are already congested, so the proposal in this paper can be presented and is a means that allows for lowering congestion to a manageable level by separating traffic into heavy (bulk) and light (pick, sync) portions and carefully controlling the former. Arguably, the two approaches are two sides of the same coin.

The following literature specifically targets *dense wireless networks* and can therefore be compared to this paper. In [4],

the author discusses the 802.11ax protocol, the release of which is planned for 2019. 802.11ax places the problem of *dense wireless spaces*, which it is defined as high density of APs (itself coming from high density of users), in the center of its new features, which include new *beamforming* techniques but, more importantly, a much higher degree of coordination between network and user sides. The proposal in this paper can be viewed as a form of such coordination where, in the absence of widespread adoption of 802.11ax, sources and destinations of wireless traffic have to coordinate in an ad hoc manner.

The work in [8] has a similar proposal but focuses on 4G+ networks, specifically the LTE-A standard. Again, [8] argues that client-network coordination is the only feasible solution to the problem of congestion in dense wireless spaces. Given the centralized nature of the LTE-A protocol, [8] advocates that, under high interference, some small cells can be turned off in order to reduce the level of interference. Note that this solution is very similar to the *Bulk-n-Pick* method in this paper, where interference is controlled by coordinating sessions of bulk transfer. Even with these similarities, the contexts between this paper and [8] are, naturally, drastically different.

At least one recent paper on the subject of *dense wireless spaces* contrasts with this paper as well as the [4, 8] above in that it views the solution in terms of a *multidimensional convex optimization* problem. Numeric simulations in [20] show that performance deteriorates on a fairly slow curve and compares with the curve that represents the case of *perfect beamforming* (i.e., 100% distinction/separation of each wireless client) in terms of the shape. This paper argues that the measurement studies explained within this paper as well as those presented in [4–8] and others offer the overwhelming amount of real-life evidence that supports the discussion in this paper.

With the above description in mind, it is necessary to distinguish this paper from the proposals in [4, 8]. Just like this paper, both [4, 8] propose a form of coordination between clients and network in purely WiFi [4] and 4G+ [8] networks. Both depend on technologies that are yet to be adopted for widespread use. This paper, on the other hand, depends on the conventional WiFi stacks and is therefore feasible under the current level of wireless technology. The unique distinction of this paper is that it measures and models throughput at *application level* which is different from that at lower levels; the traditional analysis target for current literature [4, 8]. This paper shows that this difference is substantial. Specifically, measurement shows that some HTTP requests can fail midway under high interference and have to be repeated. At application level, this translates into abnormally long *completion times* which, at lower protocol levels, may take several failed transmissions until the final successful one.

3. Proposal: The Bulk-n-Pick Method

This section provides more details on the proposed Bulk-n-Pick method and on how it compares with the traditional method.

FIGURE 1: Roles and procedures within one Bulk-n-Pick cycle. Thickness of curves represents relative difference in traffic size.

In the traditional method, data exchange in a dense wireless space happens in real time, disregarding bulk size. Large bulks are often split into smaller pieces (this paper uses 100 kbyte pieces) in order to be able to trace the status of the download, but otherwise the download process happens at the time when it is required by the application logic. This logic also describes downloads based on web sockets [21], which can be disrupted without losing all the previously received data and restarted from the current position. Note that statistically WebSockets and piece-wise downloads should result in similar performance, as far as congestion in dense wireless spaces is concerned.

The *Bulk-n-Pick method* proposes a different logic for web applications. Figure 1 shows one Bulk-n-Pick cycle. The total bulk of content is assumed to be packed into a single binary file and distributed to all the clients prior to using it; normally, the download will happen at the beginning of a Bulk-n-Pick cycle. For simplicity, the analysis further in this paper will assume that the total bulk is 100 Mbytes in size. Once the bulk is distributed to all the clients, the Bulk-n-Pick method assumes that the exchange between the server and multiple clients is reduced to *sync messages* which specify actions. For actions that need to use a portion of the previously downloaded bulk, sync messages would include the respective reference (offset and length in the binary bulk). For example, a sync message may say *show image* and specify the location of the binary data for the image in the bulk. Since most modern browsers support HTML5, the bulk can be handled in the raw binary form (Blob, ByteArray, etc.) without having to assign a meaningful structure to the entire bulk; the latter can simply be stored as a binary string.

On the other hand, *localStorage*, another major function in HTML5, is not necessary as the web applications (written in JavaScript) can keep the bulk in its runtime memory in form of a normal JavaScript variable (array, hash, etc.). Experiments show that runtime memory can easily hold several hundreds of Mbytes of data while *localStorage* on most modern browsers is restricted to 5 Mbytes in the default setting. Moreover, the nature of the bulk in the proposed web application is such that it only needs to exist for a

given short-term browser session, after which it can be safely discarded.

Note that Bulk-n-Pick method is multipurpose in nature and can go beyond the educational use described in this paper as well as earlier works in [11, 12]. For example, it can be applied to various over-the-network activities like indexing [22], where the application logic would be changed into short-term sessions, each with its own bulk and sync messages. The proposed method is applicable as long as it is implemented in a dense wireless space where devices need to exchange relatively large bulks at random times. Merging and scheduling bulk transfers at a preparation stage helps alleviate congestion at a later time.

The justification for the *Bulk-n-Pick* method is offered in Section 4 when presenting results of a measurement study within a mock wireless class. Measurement results show that separating bulk traffic from small-size sync messages is a way to control the level of interference in the dense wireless space. Note that this method requires *coordination* between client and server sides, which is a common ground between this paper and recent proposals in [4, 8].

4. The Wireless Class Dataset

The predecessor dataset in [12] used 3 APs and randomized *bulk size* (size of server's replies to HTTP requests from clients). Both parameters make it difficult to model Bulk-n-Pick situations which focus on *separation of bulk from picks* (small syncs) and expect simple wireless classes with only one AP. The new experiment and the resulting dataset explained in this section resolve these problems.

The following setup was used for the new experiment. The population of client devices consisted of 8 tablets and 12 notebooks, with no specific reason for this split except for the physical limits of hardware available at the time. All the devices were tightly packed within a small space (the area of 2 desks) to avoid the effect of distance to the only AP. Note that the dataset in [12] emulated a real class layout in which distance to AP is a nonnegligible variable. Both [12] and experiments in other literature [5]

FIGURE 2: Visualization of the dataset used as the basis for analysis in this paper. Request size is in *bytes* and request gap in *seconds*. Left-to-right columns are for *tablets only with no gap*, *all devices with 5 s gap*, and *all devices with no gap*. Rows are for *retry count* and *completion time threshold*, respectively.

points to strong dependence of data rate to distance from AP.

Each run in the new experiment used a fixed *bulk size* for each session, selected randomly from the list of 100 bytes, 10 kbytes, and 100 kbytes. The *gap* parameter was used to define the time interval between requests by each client; 5s and 0s gaps were used, representing the intensity of traffic exchange. It was assumed that the 100 Mbyte bulk, the fixed size of the total bulk of content in this paper, was to be downloaded gradually in pieces using multiple 100 kbyte downloads, while 100 byte requests were used for later syncs. The 100k pieces are necessary from the practical point of view; they allow the web application to shown to human users the current status of the download. Status can also be shown when using web sockets [21] but that method requires more programming effort without offering any functional improvement. In dense wireless spaces, continuous sockets for each client would increase congestion and would be frequently disrupted. More discussion on pieces versus sockets is offered further in this paper.

To avoid statistical bias, 10 sessions were conducted for each unique combination of the above parameters, and the average was selected as the representative value. Following the same general goals for statistical analysis as were explained in [12], experimental results are processed in such a way as to show probability of a given number of resent web

requests and a given total completion time, for each unique configuration. The term *probability* here is a purely statistical concept, easily derived from occurrence frequency of a given event.

Figure 2 visualizes the new dataset. In order to present larger plots, axes are shown only for edge plots but are the same for all plots in the visualization (however, each row has a different vertical axis).

The top row shows how many tries (repeated requests) it took to download a piece (100 bytes, 10 kbytes, and 100 kbytes) while the bottom row shows how much time in total it took to complete each request (*completion time*), including the multiple tries. The visualization logic is such that each horizontal coordinate (virtual cut) represents *the probability* for a given request to encounter the corresponding value on the vertical scale. The probability is calculated as *occurrence ratio* based on the statistics collected in experiments. The value is cumulative, which means that 0% of requests experience the value above the largest (applies to both tries and completion time) and 100% of requests experience the value above (threshold value included) the smallest in the dataset. All the values in between are produced by lowering the threshold and counting the ratio of requests that produced values above it. Annotations in the rightmost column explain a randomly selected bullet, as an illustration of the visualization logic. Note that the probability curves in Figure 2 are used as models

in analysis further in this paper. The models simply pick a given value from the dataset based on its probability and use it to emulate a given wireless classroom. Note that running such experiments in real classes is too costly and is left to future practitioners of the proposed technology.

The distinct experimental conditions that form columns in Figure 2 are as follows. The 1st column is from the experiment which used only the 8 tablets (*tabletsonly*) and zero gap between requests. This column is also the only one that shows all the three curves for individual sizes: the 10 k and 100 byte curves were found to be the same and therefore are redundant in other columns; note that this point can be considered as an experimental proof of the advantages offered by the Bulk-n-Pick method as 100-byte traffic can only be used for real time sync.

The second and 3rd columns show only the 100 k curves for the entire population of devices (8 tablets plus 12 notebooks are tagged as *all*) and 5 s and 0 s gaps, respectively.

Note that 2nd and 3rd columns are drastically different. Since only the *gap* parameter is different, it can be considered as the main factor in the congestion experienced by all the devices in the 3rd column. In the upper plot, we see that most requests are retried with about 40% of requests taking three or more retries to complete. This effect is also shown in the bottom plot in the greatly deteriorated performance in completion time. This experiment supports the general findings in existing literature [5] which argues that duration of sessions has a major effect in dense wireless spaces.

The above experiment supports the following modeling used as basis for the analysis further in this paper. The 1st column of plots and specifically its 100 k curves in Figure 2 are used directly to model the Bulk-n-Pick model because only a subset of devices participate in bulk transfers; this assumption is valid for up to 8 devices tested in the experiment. Note that 1st column shows that there are no (or very few) requests which required 2+ tries to complete, while distribution of completion time for 100 k bulk is only slightly above that for 10 k and 100-byte bulks.

The 100 k curves in the 3rd column are used to model the traditional method as all the devices share the same wireless channel and use it only for bulk traffic (no separation of bulk from sync).

More details on modeling are provided further in this paper. This dataset is used for two distinct analyses. First a simple analysis is performed directly based on this dataset. Then, the dataset is transformed into the *density-rate* space which is used for additional emulation and analysis based on the optimization problem formulated further in this paper.

5. Trace-Based Analysis

The following setup is used for analysis. First, the dataset and basic modeling method were explained in Section 4 and are used as the basis. Secondly, the analysis objective is simplified by only emulating the initial bulk download and keeping the sync part of web application sessions out of scope. This is a valid simplification as the experiment described in the previous section showed that 100 byte and even 10 kbyte message did not cause congestion even with

FIGURE 3: Example emulation run with 2 parallel sessions in the proposed Bulk-n-Pick model. Each result is plotted as distribution in increasing order of values, which also happens to be the time order.

20 devices in class. In other words, by separating bulk from pick messages and scheduling bulk downloads, the proposed method resolves the congestion problem and can assume that sync traffic is handled in normal (uncongested) conditions. For the sake of the argument, mild congestion can sometimes occur even for sync traffic, in which case sync messages might fail. However, two failures in sequence are rare, which means that resilience can be easily achieved by repeating the sync message. Note that this is different from the heavy congestion discovered in [12] where large-volume messages would have to be repeated 3–5 times and would take much longer to complete (lower rates due to congestion).

The *completion time* metric of performance refers to the completion of all bulk downloads for all the devices in class, which means that it is the sum of all individual bulk downloads. As was explained above, all the bulks are downloaded in 100 k pieces. The number of parallel sessions for the Bulk-n-Pick model is a variable in emulation but is kept within the range found in the above experiments so that analysis results remain valid.

Figure 3 shows an example plot for 2 concurrent sessions in the Bulk-n-Pick model compared to the traditional model in which all devices compete for the channel. Note that *two concurrent sessions* does not refer to the total number of downloads but is simply the specific scheduling quota which allows for at most two devices to download the bulk at the same time. The schedule is asynchronous and the next download starts soon after the previous one completes, allowing for a small gap required for control messages.

The plot shows time progress of individual completion times for each device while the last (rightmost) value on each curve is the final *completion time* as per the definition above. As can be expected, the Bulk-n-Pick downloads are fast for each pair, hence the stepwise shape of the curve, but it still takes time for the entire class to complete. Under the traditional model, all the devices complete at roughly the same time. The overall outcome is that the proposed Bulk-n-Pick method is better than the traditional method even

FIGURE 4: Overall performance plotted for an increasing array of parallel sessions. Bullets for the traditional model are copied in each bullet because its performance does not depend on the number of parallel sessions.

when only two devices are allowed to download the bulk concurrently.

Figure 4 shows the overall performance ranging from 1 to 10 concurrent devices (under Bulk-n-Pick model). The traditional model shows the same performance for each horizontal position as it does not depend on the concurrency parameter. Each bullet has the 3σ bounds marked to visualize variance of data aggregated in a given bullet. The results show that the Bulk-n-Pick method performs poorer than the traditional method only when concurrency is set to 1 but starts outperforming it from 2 and more concurrent bulk sessions. With 5 concurrent sessions, it takes about 30 m for the entire class to complete the download. With 10 devices, the total completion time is down to about 15 minutes, a reasonable preparation time for an educational class [11]. The value for 10 devices is extrapolated from the experiments for 8 tablet devices as explained in Section 4 and is therefore very close to experimental conditions.

6. Online Density Optimization

This section opens the second part of the proposed method, specifically, the method that governs the online optimization of performance under unknown (in advance) load. Note that this is a major distinction from the analysis in Section 5 where the core assumption was that bulk would be scheduled in advance. The optimization in this section removes this assumption and instead monitors congestion in real time. The control side is still present; that is, when a high level of congestion is detected, the server side can control it by decreasing the number of devices allowed to run concurrently. This level of control can be achieved with ease with existing software. For example, web server can adjust *sync interval* for all clients, thus forcing more sparse traffic conditions. Direct scheduling (e.g., “come back in 1 minute”) command can be scheduled for individual clients when bulk transfer is in question. However, note that this *scheduling* does not

mean that the entire schedule is being optimized. Rather, this *scheduling* refers to relative delays issued to individual clients in such a way that the noncongested level of traffic in the wireless space can be maintained. In this respect, although the proposed optimization has an indirect effect on the schedule of requests, the problem below is formulated as *decision rules* rather than a *scheduling problem*.

Let us assume that we have a history of n records of recent performance in the local wireless space. While the records are collected, the window of k most recent records is kept separately for real time monitoring. Let us also denote transmission rate as R , bulk volume as S , and density as D .

The *density* metric D requires special attention. It represents *intensity* of HTTP requests sent by all clients, calculated as the number of requests per unit of time. However, since the core topic of this paper is referred to in academic literature as *dense wireless space*, it was considered that the same name would be more appropriate for the metric as well. Although the *density* metric represents *throughput* (or *goodput* meaning the throughput of successful requests), it does not directly represent the physical level of congestion. Rather, it is measured at server side, which can only observe successful transfers. As was mentioned earlier in this paper, when the wireless space is congested beyond a certain level, connections carrying HTTP requests can fail midway and even result in empty replies returned to the client. The robust application logic deals with it by resending requests multiple times. Since the *density* metric is measured at the server side, it cannot capture all the failed requests and only sees those that have completed successfully. However, using *density* D , *rate* R , and *volume* S together, it is possible to clearly identify congested conditions, as done in the next section. Note that this framework is standard and shared with other literature on the subject, majority of which analyzes *throughput* as a function of *interference*.

The number of concurrent clients is denoted as m , and the constant M_{bulk} stands for the maximum possible number of concurrent clients that can transfer bulk in parallel without causing congestion; this paper will use 8, as was identified by experiments and discussed earlier in this paper. In practice, this number can vary depending on the settings. For example, in multi-AP classes or when using high-end APs, a larger number of concurrent clients can be supported.

For a practically feasible optimization logic, it is important to find the threshold beyond which conditions are considered as *congested*. Generically, let us define α as a *margin* relative to the highest values observed for R , S , and D throughout all the available history (not the recent window k but the entire history of n samples). Then, we can define the margins of *perfect performance* as

$$\begin{aligned} R_{\max} &= \max(R_i) \quad \forall D_i > \alpha D_{\max}, \quad i \in n, \\ S_{\max} &= \max(S_i) \quad \forall D_i > \alpha D_{\max}, \quad i \in n. \end{aligned} \quad (1)$$

These rules basically define boundaries uncongested operation based on threshold levels of *volume* S and *rate* R . Note that *density* D is used to identify both thresholds. This is a major condition for the success of the proposed method. It is assumed that requests come at the maximum

possible rate, referred to as *zero gap* earlier in this paper. This allows for a simple practical assumption that *density* (as defined above) decreases only in response to changes in traffic or congestion conditions, the two directly related to each other. When presenting the dataset earlier in this paper, it was pointed that performance deterioration is best represented as a step function, that is, drastically worsening shortly after a given threshold.

In the above equations for thresholds, this artifact is translated into the assumption that uncongested performance has to stay within the α margin from the best possible performance, with the latter measured in terms of request density D . The figure in the next section will revisit this discussion with a visual proof to this assumption.

Given these conditions, the online performance optimization method at the current time slot i is then formulated as the decision on whether or not to impose a limitation on the number of concurrent client devices. Assuming that M_{bulk} is the constant representing the maximum number of allowed concurrent devices (8 comes from experiments and is used from this point on), the decision rule is written as

$$m_i = \begin{cases} M_{\text{bulk}}, & \text{when } \text{avg}_{j \in k} R_j < R_{\text{max}}, \\ \infty & \text{otherwise,} \end{cases} \quad (2)$$

where ∞ stands for *unlimited*, that is, free traffic exchange by all clients at *zero gap*.

Note that the above formulation takes the form of *decision rules* based solely on *performance statistics collected in real time*. The rules apply as long as the distribution of *density* D is similar to the one described above. Here, experimental data collected for this paper is supported by several other academic papers [5–7]. Specifically, the notion of *abrupt deterioration* of performance (a kind of saturation) under high interference is shared with several recent works [4, 8, 20], all containing measurement data similar to the one presented in this paper. The thresholds in the proposed rules can be adjusted for a given practical situation by detecting the step-function point in the distribution curve, as explained above. This supports the claim that the proposed optimization problem is generic in nature. See the next section for details on how the thresholds are identified in practice.

7. Emulation Setup

Figure 5 is the same dataset as was introduced earlier in this paper but this time represented in the space of *density* and *rate*. In this form, the dataset is intended to support the optimization method in the previous section and provide the visual means of its interpretation.

The map in Figure 5 was generated in the following way. First, all measurements for *gap* above zero were discarded; this is because the proposed optimization method depends on the continuous exchange of traffic in the *zero-gap* mode. Then, the dataset was replayed in time using the window k of 3 s worth of measurement samples, with 1 s step. For each window, the selected single measurement result (the center of current window) provides the size S (0.1 k, 10 k, and

FIGURE 5: Visualization of the subset of the original dataset used for the emulation analysis in this and the next sections.

100 k as explained above), while the entire window provides the number of requests (i.e., density D) and the average rate R . The average rate is calculated as the average value across all the samples within the window, where each sample would contain its own individual rate (calculated based on the completion time of an individual request).

After manual confirmation using the raw data, the areas in Figure 5 were marked with volumes 0.1k, 10k, and 100k and, on the right side, number of devices (8, 20). There is a diagonal line which separates the noncongested from congested performance, as marked in the figure.

Let us consider how congestion increases, based on the visual proof in Figure 5. Vertical column around $D = 35$ is the area of *perfect performance*, which was found for all measurements for 0.1k and 10k traffic volumes; that is, the network would not get congested when all 20 client devices would continuously receive 10k chunks of data.

The left side of the diagonal line is fully dictated by traffic which uses 100k chunks. Here, the areas are split into two groups. Several areas at the upper range of *rate* (note that the vertical scale is in log) come from experiments with 8 concurrent devices. The rest of the areas, including those at the bottom of the range, come from experiments with 20 concurrent devices.

The map is shown as the grid pattern, where each cell is filled with a color that represents the relative occurrence frequency. Most of the areas have darker areas in the middle, which is a natural dispersion pattern which would have a central mode and some scattered samples nearby. Note that the darkest cell in the map is at near absolute zero; this is part of a 100k/20 area which was observed in extremely congested conditions, when some clients had to send several repeated requests before succeeding. This is reflected in both lower density of requests (near zero) and very low rate.

The spatial distribution of areas for 8 versus 20 devices is interesting. There is a clear visual pattern that states that, in congested conditions, by decreasing the number of concurrent devices from 20 to 8, one can not only increase the density of requests but also drastically improve (0.5–1.5

magnitudes on the log scale) transmission rates (by extension, number of retries) for individual requests.

This, in a nutshell, is the visual way to explain the proposed optimization method. First, congestion is detected when the diagonal line is crossed (α margin is a variable) and *rate* decreases below the maximum achievable rate on the right side of the diagonal. Then, by limiting the number of allowed concurrent devices, the overall performance is improved by increasing both density and rate.

Based on this general approach, emulations for analysis further in this paper were conducted as follows. Measurement samples behind the visual in Figure 5 were selected for replay based on their relative occurrence frequency. Initially, the default state assumes that 20 client devices perform traffic exchange with the server concurrently. This means that samples from measurements with 8 devices would not be selected. Otherwise, the samples are selected randomly, forming a continuous stream of requests which are monitored using the window of $k = 100$ samples.

Random selection frequently leads to conditions, when current window of samples experiences congestion. Then, based on the thresholds and decision rules explained in Section 6, the server may decide to switch to the 8-device mode. In this case, random selection of samples is limited only to the measurement data coming from experiments with 8 devices. When congestion is no longer detected, emulation switches back to the default conditions and resumes random selection of input data from the 20-device subset of the dataset.

Many simulation runs were conducted to cover a range of values for α (0.9, 0.8, 0.7, 0.6, 0.5) and remove the randomness effect. Each simulation run lasted for 5000 randomly selected samples.

8. Analysis of Emulation Results

Since this part of the analysis assumes that some actions should be taken in real time, it makes sense to give proper names to the competing methods. The *Do Nothing* method will represent the traditional approach, which does not, in fact, do anything about congestion. The *optimized density* method implements the proposed optimization problem and the visual heuristic explained in Section 7. Unfortunately, as the review of literature above showed, there are no other rivals at the time to be included into the comparison.

Figure 6 shows trajectories for performance under the *Do Nothing* (above) and the proposed (below) methods. The term *trajectory* refers to the curve that represents dynamics in *rate* with variable *density*. Data for the trajectories was filtered in such a way that the samples *beyond* the threshold (i.e., within the margins of the perfect performance) were not included in the plots. In other words, only the data, which represents the emulated *congested* conditions, is shown. Larger bullets represent larger margins α . There curves are plotted for each method, each for its own margin.

Let us deal with the easiest part of the plots first. The near-zero area in the *Do Nothing* method is the legacy from the dataset map in Figure 5. Since it is not within the margins of *perfect performance* and since the traditional method does not

FIGURE 6: Performance trajectories for the two methods of coping with dense wireless spaces. Each plot has three curves in the decreasing order of bullet size, each for the margin values $\alpha = 0.9$, $\alpha = 0.8$, and $\alpha = 0.7$, respectively.

optimize the performance in real time, this area is retained. In other words, the traditional technology is expected to keep suffering from congested wireless spaces. Otherwise, performance trajectories for the *Do Nothing* method gather around the same coordinates, with the exception of the $\alpha = 0.9$ case, when it sharply moves into the area of *density* of 30–35. This is purely due to the high (close) margin, which in this case classifies some of the samples very close to perfect performance as those under congestion.

For the proposed method, the two smaller trajectories are also very similar to each other. Both experience relatively high and stable *rate* across a wide range of values for *density*. However, when $\alpha = 0.9$, the trajectory is drastically different in that it shrinks to a relatively focused area with high values in both *rate* and *density*. This is an interesting effect that hints that it is better to react to changes in *density* early on, even with deviations at 10% from the level of *perfect performance*.

By comparison, the *Do Nothing* method is a convex function, which offers poor performance at both ends of the curve and relatively good performance only close to its peak. The proposed method, on the other hand, offers stable rates across a wide range of density conditions. The side feature of the proposed method is its relative stability to the α parameter. Based on this dataset, the value of $\alpha = 0.9$ is

recommended as it represents a monitoring routine which reacts quickly to deviations from perfect performance and thus achieves very high rates and a smaller range of density.

9. Conclusion

This paper proposed the new Bulk-n-Pick method for data transfer in dense wireless spaces. The specific case analyzed in this paper is the *one-to-many* transfers as found in educational classed (lecturer to students) but the method itself is applicable to the more general *many-to-many* case.

The core idea of the proposed method is to separate *bulk* from *sync* traffic and strive to download the bulk via a controlled schedule of concurrent sessions. Experiments in this paper show that 8 concurrent devices cause very little congestion, and the further analysis shows that with 10 concurrent devices a class of 20 clients can complete all the individual downloads in 15 minutes, while the same class would take over an hour in the traditional mode when all the devices would compete for the same channel under extreme congestion.

The analysis in this paper was divided into two parts. The first part performed simple analysis based on the raw experimental results. The second part focused on a general method for optimizing performance in real time. There, an optimization problem was formulated as a combination of *thresholds* and *decision rules* that were triggered when performance exceeded a given *margin* of deviation from the near-perfect performance. The main distinction between the two analyses is that the first one basically proposed a scheduling method, while the second one represents a generic solution to the same problem.

The optimization-centric analysis showed that the dataset could be represented as a *density-to-rate* map, which could serve as a visual guide to the steps that should be undertaken towards optimized performance. The visual also served as a proof for the decision rules formulated earlier. The data behind the visual was used for the trace-based emulation analysis, which showed that only the proposed optimization can maintain a steady average rate within the wireless space regardless of the current density (intensity) of requests. It was also found that the proposed method is advised to react swiftly even to relatively minor deviations from the perfect level of performance as such strategy would result in yet higher rates and a shorter range of observed densities.

This paper was set solely in the one-to-many setting, a good example for which is an educational class. Future publications on the subject will be generalized to many-to-many wireless environments. This will introduce new elements into the problem, some of which were briefly mentioned in this paper.

Competing Interests

The author declares that they have no competing interests.

References

- [1] IEEE Standard, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: further higher-speed

physical layer extension in the 2.4 GHz band," IEEE Standard 802.11g, supplement to part 11, 2003.

- [2] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," IEEE Standard 802.11n, 2009.
- [3] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, 2010.
- [4] B. Bellalta, "IEEE 802.11ax: high-efficiency WLANs," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.
- [5] Aerohive Networks, "Aerohive design and configuration guide: high-density WiFi networks," Aerohive Whitepaper, 2013.
- [6] B. Bing, *Emerging Technologies in Wireless LANs: Theory, Design, and Deployment*, Cambridge Press, Cambridge, UK, 2008.
- [7] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11," *IEEE Communications Letters*, vol. 6, no. 8, pp. 355–357, 2002.
- [8] B. Soret, K. Pedersen, N. T. K. Jørgensen, and V. Fernández-López, "Interference coordination for dense wireless networks," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 102–109, 2015.
- [9] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, "Beacon-stuffing: Wi-Fi without associations," in *Proceedings of the 8th IEEE Workshop on Mobile Computing Systems and Applications (HOTMOBILE '07)*, pp. 53–57, IEEE, Tucson, Ariz, USA, February 2007.
- [10] Japan Ministry of Internal Affairs and Communications, *School LAN: Layouts and Recommendations for Building Local Area Networks in Schools*, Japan Ministry of Internal Affairs and Communications, 2007.
- [11] M. Zhanikeev, T. Tomoto, and H. Watanabe, "Time to play to know (T2P2K): first report on a digital classroom project in grades 4 through 6 of elementary schools," *IEICE Technical Report on Educational Technology (ET)*, vol. 112, no. 374, pp. 37–42, 2013.
- [12] M. Zhanikeev, "Experiments on practical WLAN designs for digital classrooms," *IEICE Communications Express*, vol. 2, no. 8, pp. 352–358, 2013.
- [13] M. Zhanikeev, "Bulk-n-pick method for one-to-many data transfer in dense wireless spaces," in *Proceedings of the IEEE International Conference on Networking and Network Applications (NaNA '16)*, pp. 90–94, IEEE, Hakodate, Japan, July 2016.
- [14] R. K. Schmidt, T. Leinmüller, E. Schoch, F. Kargl, and G. Schäfer, "Exploration of adaptive beaconing for efficient intervehicle safety communication," *IEEE Network*, vol. 24, no. 1, pp. 14–19, 2010.
- [15] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular Ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.
- [16] S. Glisic, *Advanced Wireless Networks: Cognitive, Cooperative and Opportunistic 4G Technology*, John Wiley & Sons, 2009.
- [17] E. Dahlman, S. Parkvall, and J. Skold, *4G LTE LTE-Advanced for Mobile Broadband*, Academic Press, 2014.
- [18] M. Rumney, *LTE and the Evolution to 4G Wireless, Design and Measurement Challenges*, John Wiley & Sons, 2013.
- [19] C. X. Mavromoustakis, A. Bourdena, G. Mastorakis, E. Pallis, and G. Kormentzas, "An energy-aware scheme for efficient spectrum utilization in a 5G mobile cognitive radio network architecture," *Telecommunication Systems*, vol. 59, no. 1, pp. 63–75, 2015.

- [20] Y. Shi, J. Zhang, B. O'Donoghue, and K. B. Letaief, "Large-scale convex optimization for dense wireless cooperative networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4729–4743, 2015.
- [21] M. Zhanikeev, "Experiments with application throughput in a browser with full HTML5 support," *IEICE Communications Express*, vol. 2, no. 5, pp. 167–172, 2013.
- [22] M. Zhanikeev, "A new practical design for browsable over-the-network indexing," in *Proceedings of the International Conference on Information Science, Electronics and Electrical Engineering (ISEEE '14)*, pp. 1686–1690, IEEE, Sapporo, Japan, April 2014.