Investigating Predictive Model-Based Control to Achieve Reliable Consistent Multipath mmWave Communication

David A. Hayes, David Ros, Özgü Alay, Peyman Teymoori, and Tine Margretha Vista.

ARTICLE INFO

Keywords:
Proxy
mmWave
multipath
mobile networks
5G/6G/+

ABSTRACT

Millimeter-wave (mmWave) radio is a key building block in 5G and beyond cellular networks. However, mmWave channels are very sensitive to environmental conditions and depend on Line-of-Sight connections to provide very high data rates. Achieving *reliable, consistent communication* — i.e., a steady link rate together with low delay — over mmWave links is therefore a challenging problem. The goal of this work is to explore the use of *predictive control* to manage and simultaneously use multiple available mmWave paths to achieve reliable consistent communication by means of a multipath proxy. We investigate transient solutions of Markov Modulated Fluid Queues (MMFQ) to model the short-term evolution of the proxy's packet queue, consistent with the use of Markovian models to capture the behavior of mmWave channel blocking. We propose a combination of models that can be solved using newly proposed matrix-analytic techniques in a timely enough manner for use in real-time control. This gives us a prediction, over a short time horizon, of either proxy queue distributions or probabilities of reaching particular proxy buffer levels. Thus, it enables the proxy to make *preemptive* path decisions in order to maintain a desired Quality of Service. A proof-of-concept simulation study demonstrates the efficacy of our proposed MMFQ-based predictive approach over both static and purely reactive control approaches. Further, we explore the potential benefits of a hybrid approach to path management, combining both predictive and reactive control. This can allow the controller to cater for unforeseen events that cannot be forecast by the predictive controller, mitigating the resulting extra queuing and corresponding delay spikes.

The published journal article is available here:

https://doi.org/10.1016/j.comcom.2022.07.011

https://www.sciencedirect.com/science/article/pii/S0140366422002596

Computer Communications, Volume 194, 2022, Pages 29-43, ISSN 0140-3664,
Part of the ACM MSWiM 2021 Special Issue

Investigating Predictive Model-Based Control to Achieve Reliable Consistent Multipath mmWave Communication

David A. Hayes[a,*], David Ros[b], Özgü Alay[c], Peyman Teymoori[c] and Tine Margretha Vister[c]

[a] *SimulaMet, Oslo, Norway*
[b] *Simula Research Laboratory, Oslo, Norway*
[c] *University of Oslo Norway*

ABSTRACT

Millimeter-wave (mmWave) radio is a key building block in 5G and beyond cellular networks. However, mmWave channels are very sensitive to environmental conditions and depend on Line-of-Sight connections to provide very high data rates. Achieving *reliable, consistent communication* — i.e., a steady link rate together with low delay — over mmWave links is therefore a challenging problem. The goal of this work is to explore the use of *predictive control* to manage and simultaneously use multiple available mmWave paths to achieve reliable consistent communication by means of a multipath proxy. We investigate transient solutions of Markov Modulated Fluid Queues (MMFQ) to model the short-term evolution of the proxy's packet queue, consistent with the use of Markovian models to capture the behavior of mmWave channel blocking. We propose a combination of models that can be solved using newly proposed matrix-analytic techniques in a timely enough manner for use in real-time control. This gives us a prediction, over a short time horizon, of either proxy queue distributions or probabilities of reaching particular proxy buffer levels. Thus, it enables the proxy to make *preemptive* path decisions in order to maintain a desired Quality of Service. A proof-of-concept simulation study demonstrates the efficacy of our proposed MMFQ-based predictive approach over both static and purely reactive control approaches. Further, we explore the potential benefits of a hybrid approach to path management, combining both predictive and reactive control. This can allow the controller to cater for unforeseen events that cannot be forecast by the predictive controller, mitigating the resulting extra queuing and corresponding delay spikes.

1. Introduction

Reliable consistent communication is a fundamental requirement for a range of upcoming network applications like extended reality and advanced emergency communications. These applications, based on interactive high-definition video, are expected to be important use cases for future cellular networks. However, they necessitate the network to provide very high bit rates that are as stable and consistent as possible. At the same time, their interactive nature makes them latency-sensitive, so they benefit from stable low delays.

Millimeter-wave (mmWave) radio, in the form of 5G New Radio (NR) access, is an important addition to the set of new technologies deployed in 5G cellular systems. It is also expected to play an even more salient role in 6G networks and beyond. Wireless links operating in the 28 GHz band or above offer much higher data rates (e.g., several Gbps) than those available with traditional frequency ranges used in older cellular systems. However, radio propagation in these frequency bands is highly sensitive to atmospheric conditions like rain and water vapour [33]. Line-of-Sight (LoS) propagation is required to achieve the very high data rates, given that mmWave is easily blocked by walls, foliage and people, thereby introducing huge fluctuations in the data rates [38].

Achieving stable and reliable consistent communication over mmWave radio links with such inherent characteristics is therefore a difficult problem. Coming mobile systems, i.e., 6G and beyond, will rely even more than 5G systems on mmWave links, and at even higher frequencies than today, further exacerbating the issue. This makes the problem even more important to solve, if the promise of 6G and beyond to support high-definition, interactive video applications is to be realized.

It is well known that wireless links in general, and mmWave links in particular, may degrade the performance of transport protocols such as TCP [49, 32]. For this reason, performance-enhancement proxies (PEPs), and in particular TCP splitting PEPs (SPEPs), are widely deployed in cellular networks [50] to mitigate the impact of wireless links on transport protocols. SPEPs optimize data transfer over the wireless hop for specific applications (e.g., web browsing). In [17] we showed how existing carrier SPEPs could provide performance benefits for mmWave-like channel dynamics. However, such proxies were designed for and deployed on 4G networks, operating over a single radio channel.

A complementary approach to provide reliability and performance improvements in wireless links is the use of multipath transport protocols. These have seen not only significant research efforts but also wide-scale, real-world deployment [6, 5]. In 5G, the proposed Access Traffic Steering, Switching, and Splitting (ATSSS) architecture [1] provides

davidh@simula.no (D.A. Hayes); dros@simula.no (D. Ros); ozgua@ifi.uio.no (Ö. Alay); peymant@ifi.uio.no (P. Teymoori); tinemv@ifi.uio.no (T.M. Vister)

ORCID(s): 0000-0002-1122-1306 (D.A. Hayes); 0000-0002-6121-3494 (D. Ros); 0000-0001-5800-2779 (Ö. Alay); 0000-0002-9507-4373 (P. Teymoori); 0000-0002-3852-9159 (T.M. Vister)

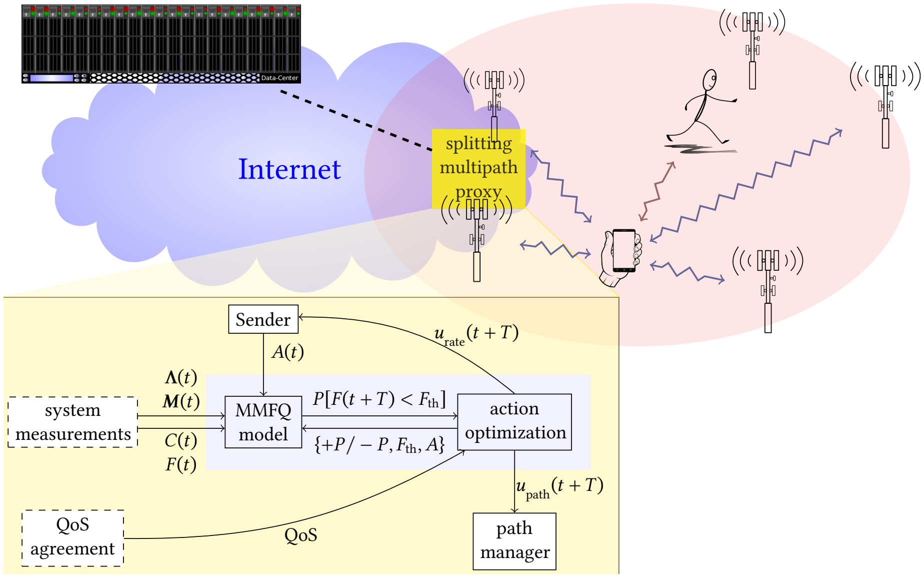


Figure 1: System architecture overview of mmWave scenario with five base stations connected to the multipath proxy. For the **MMFQ model** block see § 3. For investigations into MMFQ model-based control of mmWave paths (u_{path}) to maintain good QoS, see § 5. Future work describing **action optimization** where constraints cannot be met without also controlling send rates (u_{rate}) is described in 6.3.[2]

support in the 5G core for transport layer multi-connectivity between 3GPP and non-3GPP networks. ATSSS is planned to be expanded where multi-connectivity is also leveraged between multiple 3GPP networks. Given dense small-cell deployments of mmWave links[1], we argue multipath transport will play a key role to better cope with mmWave link impairments. Some preliminary work in this direction [37] shows the advantage of using multipath transport for combining mid-band and mmWave links in LoS and non-line-of-sight (NLoS) scenarios, ultimately showcasing LTE/5G and WLAN multi-connectivity.

In this paper, we study how *multipath proxies* could be used to leverage multiple parallel mmWave links to satisfy demanding Quality of Service (QoS) requirements in terms of stable, high data rates *and* low delay. The problem we are trying to solve is *not* that of a "greedy" data source, but rather that of communicating reliably at a particular (high) rate without large queuing delays being introduced by the multipath gateway. We envisage a scenario similar to Fig. 1 where a mobile device is equipped with a radio with multiple mmWave connectivity support, so the mobile device can simultaneously connect to multiple base stations (BSs). Considering the LoS path to a BS may be temporarily blocked due to movement of the mobile device or objects around it, our goal is to dynamically select the minimum number of mmWave links necessary[3] to provide the required QoS, for a given application and User Equipment (UE). We propose the use of a *splitting multipath proxy* that separates the multipath mmWave domain from the Internet with two key roles: (i) for a given application and UE, select the minimum number of mmWave links necessary to provide the required QoS, and (ii) schedule data packets for transmission across the different selected links. In this paper, we will focus solely on the *multipath management* role (i). The scheduling role (ii) is ongoing work that will be discussed in § 6. The issue of how to balance network and other costs with QoS requirements by means of the full control system depicted in Fig. 1 is also discussed in § 6.

Our contributions in this paper are threefold:

1. In § 3, we develop relatively simple *mathematical models* that try to capture the LoS/NLoS dynamics of a set of links and the resulting, aggregate bit rates that allow to drain an application flow's buffer in the proxy. The goal is for the proxy to be able to predict, over short time horizons (say, a couple of hundred milliseconds), the state of the buffer with reasonable accuracy.

2. In § 4, we evaluate the efficiency of these models for real-time control in terms of accuracy (i.e., how accurately the models predict) and performance (i.e., how computationally complex different models are).

3. In § 5, we sketch a proof-of-concept *predictive multipath mmWave proxy mechanism* that allows the proxy to do multipath management based on the model's predictions. We test two different prediction methods: (i) a predictor of the distribution of queue level at the proxy, (ii) a predictor of the earliest time when the

[1] https://www.fiercewireless.com/5g/real-world-deployments-mmwave-5g-will-require-very-very-dense-networks-report

[2] Fig. 1 uses openclipart CC0. `https://openclipart.org/detail/`: 297779/smartphone-in-hand, 229372/mixedantenna-cell-tower, 91519/al-running,23263/datacenter

[3] Selecting the minimum number of links is motivated by energy constraints both in the UE and on the network side, as well as cost constraints on the use of multiple links.

proxy queue will reach a given threshold. Through event-based simulations, we show the effectiveness of the proposed mechanisms compared with static numbers of paths and a simple reactive method. We also explore how augmenting the purely predictive controls with a reactive method may improve the performance in some circumstances.

This paper extends our prior work in [18] as follows:

- We provide an updated and more detailed assessment of the feasibility of solving the proposed models in real time, for path management purposes (§ 4).
- We update the calculation cost experiments to reflect a more recent stable version of Julia (§ 4).
- We update one of the two basic, proof-of-concept predictive control algorithms, i.e., first-passage time based control, to better consider the case in which removal of a path may cause too large a variation of aggregate capacity (§ 5).
- We provide a more in-depth evaluation of the impact of the different path management algorithms on packet delay (§ 5).
- We introduce a hybrid control algorithm that integrates reactive control into either of the two proposed predictive control methods, and we compare its performance with that of the basic algorithms (§ 5).
- We extend the discussion on the role of the complete control system on balancing QoS with network costs, and add a brief discussion on the requirements for a scheduler in the proxy system (§ 6).

2. Related Work

Matrix Analytic: Our work applies theory from transient analysis of Markov Modulated Fluid Queues (MMFQ) to design a predictive control based on a stochastic model. Transient analysis of MMFQ is not a new field [27], however, it is only recently that advances in numerical methods have enabled solutions timely enough for it to be used for network control purposes. Work by Sericola [43] on numerical solutions to the underlying partial differential equations of the fluid queue has been a benchmark. Leveraging steady state matrix analytic solutions to MMFQs [4], there are pure matrix analytic solutions [2, 41]. However, using phase-type distributions (Erlang in this case) to give an accurate time horizon results in very large matrices. A set of matrix-exponential distributions have been proposed allowing much smaller matrices for an equivalent time horizon accuracy [19]. Recent work by Akar et al. [3] has built on this to allow efficient numerical solutions, timely enough for control purposes.

mmWave Communications' Impact on Transport Protocols: Narayanan et al. [34] present measurements of throughput, latency, application performance, and handover operations, in four different US operators' networks, three of them employing Non-Stand Alone deployment with mmWave 5G cells. Mohebi et al. [33] present a measurement campaign to investigate the performance of a 5G mmWave cell in terms of the signal and beam coverage map of an operational network, considering human body blockage effects, foliage-caused and rain-induced attenuation, and water surface effects. These studies illustrate the sensitivity of mmWave links to environment changes resulting in wide fluctuations of their capacity.

Ren et al. [42] report that TCP performance over mmWave is seriously impaired by drastic channel changes between LoS and NLoS. Poorzare and Calveras Augé [39] also analyze TCP's behavior over 5G millimeter-wave when used in a city. The authors investigate the impact of different parameters such as remote servers, RLC buffer size, different congestion control algorithms, and maximum segment size. Their results revealed that TCP could benefit from an edge server deployment due to the shorter control loop. While some approaches try to solve these issues by changing the TCP mechanisms, adjusting the sending rate intelligently to prevent degradation due to blockage [40], several methods have been proposed to use some type of TCP proxy that considers the properties of mmWave channels [25, 24, 38]. Note that all these proposals are TCP-centric, seeking to use all the available capacity, whereas our work is not specific for, nor tailored to TCP, and seeks instead to maintain reliable consistent rates.

Multipath in 5G and beyond: The benefits of multi-connectivity with support of multipath transport protocols [10, 11, 35] have motivated multipath adoption in 5G. An extensive survey by Wu et al. [47] reviews multipath transport protocols in depth, covering four core functionalities, i.e., path management, scheduling, congestion control and reliable transfer, and discusses the integration of multipath transport into ATSSS to satisfy eMBB and URLLC service requirements. There are two main approaches to 5G multi-connectivity via multipath transport solutions: *Above-the-Core* and *Core-Centric*. In Above-the-Core integration, the multipath transport protocol is deployed at both client and server sides, and the aggregation of different paths occurs in between, without impacting the network. One example of such an approach is evaluated by Wu et al. [46], where the authors consider the throughput performance for heterogeneous links including 5G mmWave links. Khan et al. [23] consider multipath on smartphones equipped with mmWave radios and evaluate MPTCP's performance in terms of power consumption. In Core-Centric integration, the multipath transport protocol is deployed at the client and in the 5G Core (i.e., through a multipath proxy), and a single-path transport is used between the core network and the server. As highlighted by several use cases [12, 22], Core-Centric integration is a stronger candidate to be adopted by 5G, since it enables more direct control of multi-connectivity within the cellular system. Our proposal, though not necessarily tied to 5G specificities, fits with the latter core-centric approach.

Multipath Path Management: This could consist in choosing the best available path for the circumstances, such as handover management where only one active path is used for transmission while other paths are used for backup (e.g., [36, 44]). Path termination is considered in several proposals, by taking into account e.g. in-order packet delivery [15], Round Trip Time (RTT) differences between paths [21], or MAC-layer information about a link's status [28]. Or, in our case, using multiple paths simultaneously as part of the same connection.

Multipath transports adopt different strategies for path management. With MPQUIC [30, 9], both hosts can negotiate multipath capabilities during the handshake, to set the state of and preferences for paths. MPTCP follows a different approach, with pre-defined path management implementations that are selected by system configuration, with some more suitable for specific environments (e.g., a *full-mesh* between all possible combinations of IP addresses of the two endpoints, to support applications that aim at load balancing or at improving throughput).

Rather than just make better use of mmWave capacity, we seek to manage paths in such a way that additional paths are set up and torn down with the aim of maintaining a consistent reliable rate.

3. Modelling for predictive control

In this work we are looking at the scenario of a mobile real-time interactive application. In particular, an application that needs a *consistent* and *reliable very high data rate* with a *very low delay*. In our scenario, the data rates needed over the radio interface require the capacities provided by mmWave. Example applications include immersive 3D video, ultra-high definition (UHD) augmented reality, etc. One use case is that of emergency responders requiring real-time UHD video augmentation of the surroundings in order to safely carry out their tasks. In such a use case, reliable consistent communication is critical. In this paper, we look specifically at the communication over the last radio hop, in particular the issue of how to make a reliable communication link out of a set of inherently unreliable mmWave channels.

The key dynamic affecting QoS over the mmWave radio link is the LoS/NLoS blocking dynamic. Paths from the user device to the mmWave base station can be blocked due to movement of the user device with respect to objects or people around them, as depicted in Fig. 1, as well as the movement of people and other dynamic objects such as vehicles. A blocked (NLoS) path may have a capacity of less than 1% of a direct LoS path. At Gbps rates, a sudden substantial drop in capacity will result in the queue proceeding the mmWave communication link to rapidly grow; resulting in high delays and data loss in just a few milliseconds. The sudden effect of high delays, high loss, and a substantially reduced capacity on our targeted applications would make them unusable (and potentially harmful in the emergency use case). The dramatic nature of the capacity changes, and the very rapid detrimental effect to the target applications, suggests a predictive approach will be needed to make system adjustments before QoS is affected. To this end we model the effect these LoS/NLoS changes have on the proxy queue over a short time horizon into the future.

Markov models have been used for many years to model human movement [26], and in particular mmWave blocking [31]. We base our work on a two-state LoS/NLoS continuous time Markov model for each path, driving the rate at which a fluid queue can be emptied, i.e., the fluid queue empties at a different rate depending on the state of the Markov model—a Markov Modulated Fluid Queue (MMFQ). The model could be expanded in the future, if more complex dynamics are observed, by introducing further states or introducing second-order characteristics to the fluid-queue drain rate [3]. However, to date we have found a first-order MMFQ sufficient for the short-term predictions used in a multipath proxy control system for maintaining a requested QoS.

We proceed by outlining the concept of a MMFQ model for control in this context, then investigate various ways of modelling the multipath system state, and evaluate them for accuracy and efficiency.

3.1. Markov Modulated Fluid Queue models for predictive multipath proxy control

The Markov part of the MMFQ models the NLoS↔LoS transitions of the various paths, and a fluid part models the queueing dynamics that result from such transitions. This can then be used to predict future queue distributions as well as the probabilities of the queue crossing thresholds within a time interval (i.e., first passage times).

The system Markov model is described by a set of states $\boldsymbol{N} = \{v_1 \dots v_N\}$, where each state v_n represents some combination of the possible NLoS/LoS path-states, and the rates of moving between these states. The rate of moving from NLoS to LoS for path k is λ_k, and the rate of moving from LoS to NLoS for path k is μ_k. For a system of K paths the set of rates is given by $\boldsymbol{\Lambda} = \{\lambda_1, \lambda_2, \dots, \lambda_K\}$ and $\boldsymbol{M} = \{\mu_1, \mu_2, \dots, \mu_K\}$ respectively, and are inputs to the system model (see Fig. 1). The rates of moving between model-states, $\boldsymbol{N}$, is the combination of $\boldsymbol{\Lambda}$ and $\boldsymbol{M}$ appropriate for the particular combination of path-states that make up a model-state v_n. This is represented by the continuous time transition matrix $\boldsymbol{Q} \in \mathbb{R}^{(N+1)\times(N+1)}$, where element Q_{ij} is the rate of moving from v_i to v_j $\forall j \neq i$, and $Q_{ii} = -\sum_{i,\forall j \neq i} Q_{ij}$. In a deployed control system, these values are based on current measurements and/or historical data collected by the network operator.

The fluid part of the model is described by the rate at which the queue can drain, i.e., the difference between the sender's rate and the available capacity (which depends on the LoS/NLoS state of the available paths). For each path, we measure the capacities[4], but in both NLoS and

[4] A second order system would also use variance in the average capacities, ($S = \{(S^{(1)}_{\text{NLoS}}, S^{(1)}_{\text{LoS}}), \dots, (S^{(K)}_{\text{NLoS}}, S^{(K)}_{\text{LoS}})\}$). For the short predictive time horizons used for controlling the system, channels are unlikely to experience enough variation to see enough benefit from using the variance compared to the extra time cost in solving it.

LoS: $\boldsymbol{C} = \{(C^{(1)}_{\text{NLoS}}, C^{(1)}_{\text{LoS}}), \ldots, (C^{(K)}_{\text{NLoS}}, C^{(K)}_{\text{LoS}})\}$. This is represented by a diagonal fluid rate (or fluid drift) matrix $\boldsymbol{R} \in \mathbb{R}^{(N+1)\times(N+1)}$, where element $\boldsymbol{R}_{ii}$ is the net rate of fluid flow into the queue when we are in the system model-state v_i. $\boldsymbol{R}_{ii} = A - \boldsymbol{D}_i$, where A is the sender's transmission rate, and $\boldsymbol{D}_i = \sum C_p$ the combined capacity for every path according to their NLoS/LoS path-state represented in system-state v_i. When $\boldsymbol{R}_{ii} > 0$, fluid is filling the queue at rate $\boldsymbol{R}_{ii}$ while in system-state v_i, and when $\boldsymbol{R}_{ii} < 0$ fluid is draining from the queue at rate $\boldsymbol{R}_{ii}$ [5].

The evolution of the MMFQ process is described by the variable $\boldsymbol{X}(t) = (F(t), N(t))$, where $N(t)$ is the state of the modulating Markov process at time t, and $F(t)$ is the fluid level at time t. $F(t)$ is limited by empty and full conditions ($0 \leq F(t) \leq F^{\max}$). The fluid queue then evolves as follows:

$$F(t + \Delta t) = \left[F(t) + \frac{A(t) - D(t)}{\Delta t} \right]_0^{F^{\max}} ,$$

where $D(t)$ depends on the Markov state $N(t)$, and $A(t)$ is the sender's transmission rate.

The transient solution to the MMFQ gives us a probabilistic prediction over some time horizon, T, into the future. Akar et al. [3] give a good description of how the solution can be obtained. Briefly, the time horizon is added by augmenting the MMFQ model with a Markovian process estimate of the time horizon to give an auxiliary model, MMFQ′. The MMFQ′ model evolves from its starting point until the time horizon is reached and then forced back to its starting point; both the fluid queue, by adjusting the drift, and the Markov process driving it. After an exponential delay, this continues forever. The steady state solution of MMFQ′ is then the transient solution over T of the original MMFQ. The transient First Passage Time (FPT) probabilities within T extend this idea, making the target threshold an absorbing barrier similar to the empty and full barriers, but allowing the system to stay at the target threshold for an exponential duration.

Traditionally the Markovian estimate of the time horizon has been a L level Phase type representation of an Erlang distribution [41]. However, the resulting state space can be significantly reduced by using a matrix exponential estimate ([19], see § 3.2.6). We solve the models using our Julia[6] implementation of the matrix analytic methods described in [3]. The model is primed with the current state of the real mmWave proxy system we are modelling: the level of fluid in the queue $F(t) = a$ (corresponding to the actual proxy queue), and underlying current state in the Markov model state $N(t) = \xi$ (corresponding to the state of currently used mmWave paths). The solution allows us to calculate the following:

1. CDFs of the buffer level for a given time horizon T, when starting at a while in state ξ:

$$P[F(t + T) \leq x \mid F(t) = a, N(t) = \xi]. \qquad (1)$$

2. The probability that the first passage time is $< T$, i.e., the chance of the queue reaching b within T when starting at a while in state ξ:

$$P[\inf_x \{F(t + x) = b \mid F(t) = a, N(t) = \xi\} < T]. \quad (2)$$

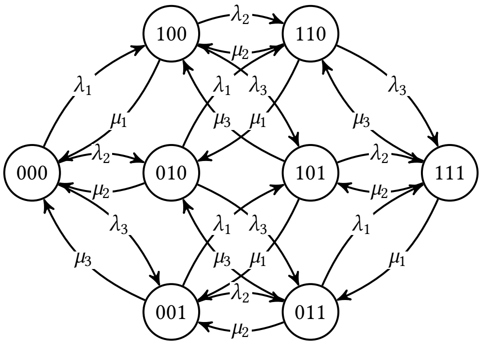


Figure 2: Full state (FULL) Markov model of LoS/NLoS state for a three base station (bs) scenario. Node labels indicate 1-LoS/0-NLoS state to each bs. λ_i is the rate of bs i changing from NLoS to LoS, and μ_i from LoS to NLoS.

Item 1 allows us to predict the future buffer occupancy characteristics and compare them to the QoS target. Item 2 allows us to estimate the probabilities of the queue level crossing certain thresholds within particular time horizons. Together they allow us to apply controls to circumvent or mitigate predicted QoS degradation before it happens, and thus maintain the desired QoS.

3.2. Predictive system performance models

Key to using MMFQs for predictive control of the envisaged reliable mmWave transport system is being able to solve the MMFQ fast enough, and to a suitable accuracy, for the prediction to be useful. In this section we investigate a number of potential models and their ability to provide feasible accurate predictions.

3.2.1. Full state Markov model (FULL)

If we consider the blockages for the different paths to be random according to a Poisson distribution, independent of each other (i.e., blockages on one path have no bearing on the blockages of other paths), and with exponentially distributed durations the scenario can be modelled as a Markov model. It may be in some real scenario that blockages are not completely independent, e.g., two physically close base stations may be blocked by the same object, or blockage of one path may ensure another path is not blocked. Still, we suggest that independence is a reasonable assumption for a well designed infrastructure.

Fig. 2 depicts a full path state Markov model for the NLoS/LoS dynamics of a 3-path system. In this model λ_i represents the rate of path i moving from NLoS to LoS, and μ_i represents the rate of path i moving from LoS to NLoS.

[5] A corresponding diagonal variance matrix $\boldsymbol{S} \in \mathbb{R}^{(N+1)\times(N+1)}$ can also be constructed, though in this paper we look only at a first-order MMFQ.

[6] https://julialang.org/. The code for the model can be found at: https://www.simula.no/file/solversrccodetgz-0/download.

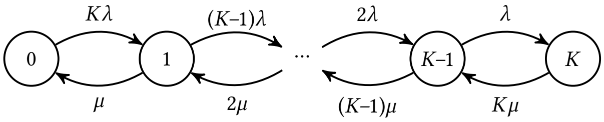


Figure 3: M/M/K/K (MMKK) model for K base stations (i.e. K paths), where each state represents the number of LoS paths.

The state-space for such a system is 2^K, where K is the number of available paths in the system. In scenarios where K is large, this model will become unwieldy and difficult to solve in the time constraints of the control system we wish to apply it to (see Table 3).

3.2.2. *M/M/K/K model (MMKK)*

If the blocking rates were not only independent but similar, and the capacities for each path were also similar, then the scenario could be modelled more simply as a M/M/K/K queue, as shown in Fig. 3. This model assumes that $\lambda_i \approx \lambda_j$, $\mu_i \approx \mu_j$, $C^{(i)}_{\text{NLoS}} \approx C^{(j)}_{\text{NLoS}}$, and $C^{(i)}_{\text{LoS}} \approx C^{(j)}_{\text{LoS}}$ for all i, j. This is a much more tractable model, even when K is large. However, even if the LoS/NLoS dynamic assumption roughly holds, it is likely paths have quite different capacities.

3.2.3. *Weighted Queue model (WQ)*

WQ relaxes the similar rates and capacities assumption of MMKK a little, by weighting the combinations of rates and capacities according to the probabilities of being in a particular state in FULL. This provides a better approximation when path LoS/NLoS change rates and capacities are different. Fig. 4 depicts the WQ model where the summary state variables are calculated as follows:

$$\boldsymbol{W}_k = \left\{ w^{(\phi)}_{\text{LoS}} \left(1 - w^{(\phi)}_{\text{NLoS}}\right), \phi \in \Phi_k \right\}, \qquad \acute{\boldsymbol{W}}_k = \frac{\boldsymbol{W}_k}{\sum \boldsymbol{W}_k} \tag{3}$$

$$\boldsymbol{\Lambda}_k = \left\{ \sum \lambda_\phi, \phi \in \Phi_k \right\}, \qquad \bar{\lambda}_k = \sum (\acute{\boldsymbol{W}}_{k-1} \cdot \boldsymbol{\Lambda}_{k-1}) \tag{4}$$

$$\boldsymbol{M}_k = \left\{ \sum \mu_\phi, \phi \in \Phi_k \right\}, \qquad \bar{\mu}_k = \sum (\acute{\boldsymbol{W}}_k \cdot \boldsymbol{M}_k) \tag{5}$$

$$\mathbb{C}_k = \left\{ \left(\sum C^{(\phi)}_{\text{LoS}}, \sum C^{(\phi)}_{\text{NLoS}} \right), \phi \in \Phi_k \right\}, \boldsymbol{C}_k = \sum (\acute{\boldsymbol{W}}_k \cdot \mathbb{C}_k) \tag{6}$$

where k is the number of LoS paths, Φ_k is the set of FULL states that have k LoS paths, and $w = \lambda/(\lambda + \mu)$ for each path.

3.2.4. *Weighted queue with additional starting states (WQx)*

For control purposes, we are particularly interested in the short term (e.g. ∼ 200 ms) transient dynamics of the buffer occupancy rather than the long term steady state. In a heterogeneous scenario where LoS rates vary greatly and the blocking rates of the available paths are also different, the current state of the mmWave system significantly influences the short term dynamics of the system. We therefore propose augmenting the simple WQ model to include the specific starting state, ξ, of the system from the full model, and perhaps a limited number of initial transitions. Fig. 4 illustrates this enhancement of WQ with one additional hop, $\xi-1$ and $\xi+1$, either side of the starting state ξ. Note that the transition rates (μ_i, λ_i) and capacities for $i = \{\xi - 1, \xi, \xi + 1\}$ are calculated in a similar manner than the WQ rates taking into account the probabilities of being in the full-model states that have been condensed. The benefit of having additional hops decreases quickly as the possible alternatives being summarised increases, resulting in states very similar to the weighted model this augments.

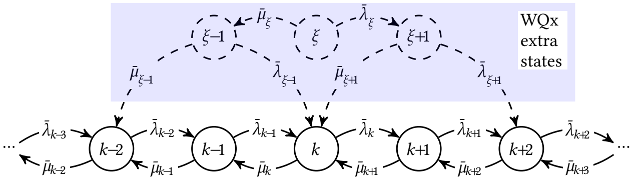


Figure 4: Weighted model (WQ) and the extended WQx model which includes an additional starting state and one jump either side (if needed).

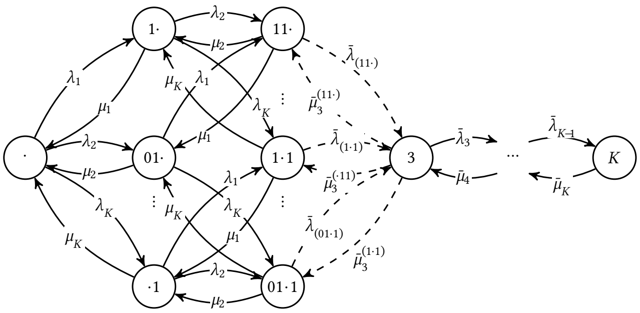

·
1·
01·
·1
11·
1·1
01·1
3
…
K
λ_1
μ_1
λ_2
μ_2
λ_K
μ_K
$\bar{\lambda}_{(11\cdot)}$
$\bar{\mu}_3^{(11\cdot)}$
$\bar{\lambda}_{(1\cdot1)}$
$\bar{\mu}_3^{(\cdot11)}$
$\bar{\lambda}_{(01\cdot1)}$
$\bar{\mu}_3^{(1\cdot1)}$
$\bar{\lambda}_3$
$\bar{\mu}_4$
$\bar{\lambda}_{K-1}$
$\bar{\mu}_K$

Figure 5: Example $K/2$ Hybrid FULL/WQx Markov model (Hybrid-2). System with $K > 3$ parallel paths, FULL used for states up to $J = 2$ with combinations of the remaining $K - J$ paths modelled collectively as WQ (starts in FULL).

3.2.5. *Hybrid FULL/WQx model (Hybrid-J)*

A summary queueing model cannot capture the nuances of particular states. Combining FULL states with the same number of LoS paths, as we do in WQ, works well if there are enough LoS paths being summed so that the differences in combined capacities is small. When the number of concurrent LoS paths represented by a state in FULL is small, the differences between the combined capacities can be large, making the WQ model inaccurate. We propose a hybrid model, part FULL and part WQx. This fully models the system when there are a small number of paths in LoS ($J = 1$ or 2 paths in LoS), but compresses the state space when there are higher numbers of concurrent LoS paths. Fig. 5 illustrates this hybrid Markov model for $J = 2$, i.e., modelling the full state for two concurrent LoS paths, and using the WQ model for larger numbers (i.e., start state is in the FULL part). Note that the state space of this type of model increases dramatically with higher values of J and K.

Table 1
Model characteristics for Figs. 6 to 9

Λ	LinRange$(0.1, 1.9, K) \times 3$
M	LinRange$(1.9, 0.1, K) \times 3$
C_{LoS}	LinRange$(0.1, 1.9, K) \times 10$ units per second
C_{NLoS}	$0.01 C_{\text{LoS}}$
buffer size	10 units

3.2.6. *Time horizon accuracy and state space*

The accuracy of using matrix analytic methods to calculate probabilities with respect to a time horizon, depends on how accurately the time horizon can be represented. The typical way of approximating a discrete time interval is by using an L-state Erlang distribution, where the higher the number of states (L), the more focused the estimate. However, the more states, the larger the resulting matrices and the longer the time taken to solve the model, potentially rendering it useless for control purposes. Horváth et al. [19] find concentrated matrix exponential (CME) equivalents to the Erlang distribution allowing similarly accurate solutions with a much smaller state space. For an Erlang distribution, the squared coefficient of variation (SCV, a measure of how focused the time estimate is) is SCV $= 1/L$, requiring very high L for a focused enough estimate of the time horizon. On the other hand, for a matrix exponential we have SCV $< 2/L^2$ for odd L. That is, for a similar value of the SCV, a matrix exponential allows the use of much smaller and more tractable matrices than those with an Erlang distribution.

The processing time involved in solving this system is related to the size of the matrices that model the scenario (the Markov state space) and the accuracy of the horizon time estimate (proportional to L). So for each type of model:

- FULL – matrix of order $2^K L$
- MMKK – matrix of order $(K + 1)L$
- WQ – matrix of order $(K + 1)L$
- WQx – matrix of order $(K + 1 + S)L$, where S is the number of additional starting states.
- Hybrid-J – matrix of order $(\sum_{i=0}^{i=J} \binom{K}{i} + K - J + \Xi)L$, where Ξ are additional start states if they occur after FULL.

K is fixed by the number of paths. Higher L gives more accurate results, but at the expense of computation time. For the hybrid models, the compromise is between J and L to achieve the best accuracy for the same computational cost.

4. Evaluation for real-time control

We evaluate our models for use in real-time path control. We look at how L, starting state, and time horizon affect the accuracy and usefulness of the models in terms of the transient queue CDF. We then look at the usefulness of first passage times with respect to the time horizon.

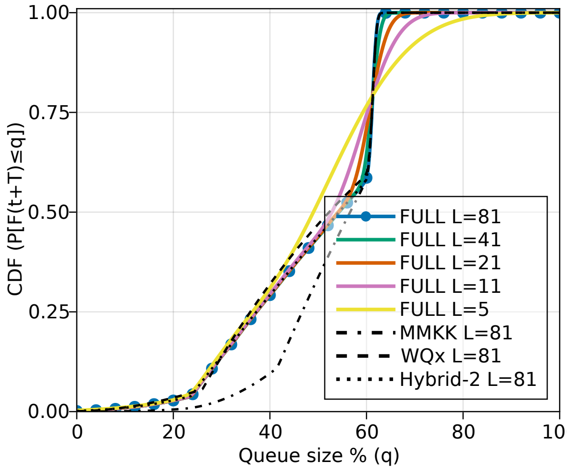


Figure 6: Relative accuracy: FULL for various T accuracies (CME levels) and condensed models. CDF over $T = 0.2$ s, buffer starting at 20%, $N(0) = 00000$ (all NLoS), $K = 5$, and load=60%.

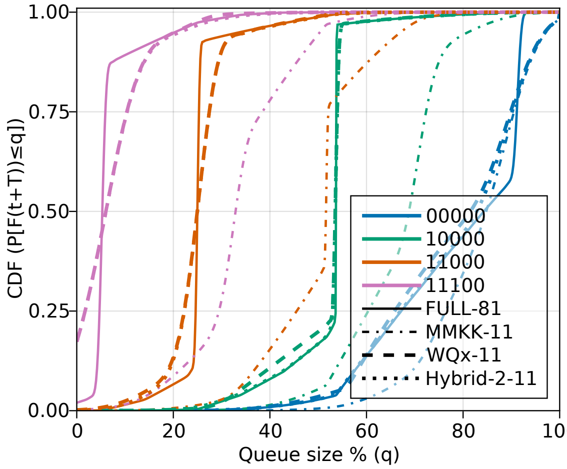


Figure 7: Impact of the starting state on the queue distribution. This is for a scenario where $K = 5$, $L = \{81, 11\}$, load=60%, and $T = 0.2$ s.

The selection of L is important since it affects the accuracy of the result as well as the time taken to calculate it. If calculations take too long, real-time control is not possible. Ideally L should be as small as accuracy constraints allow. Fig. 6 shows the queue CDF for a time horizon of $T = 0.2$ s. The transition rates and fluid flow rates are shown in Table 1. There are 5 possible paths, and the system starts with the buffer at 20% of capacity and all paths in NLoS, so the queue will initially grow. This specific scenario shows a small probability of the queue falling below 20% in 0.2 s and that it cannot get above about 60% in 0.2 s with the inflow rate of about 21 units per second (60% load). Notice that with more accurate time horizons (higher L) the model can better track sharp changes in the CDF. In this scenario, Hybrid-2 gives a CDF that is almost indistinguishable from the full model, mainly due to the system starting in all NLoS. MMKK has

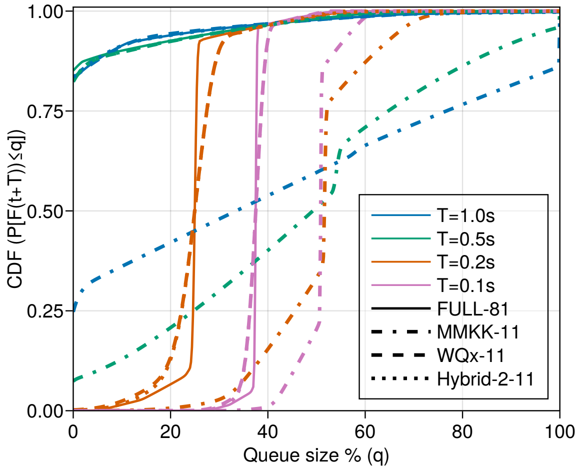


Figure 8: Impact of the time horizon on the queue distribution. This is for a scenario where $K = 5$, $L = \{81, 11\}$, load=60%, and $N(0) = 11000$.

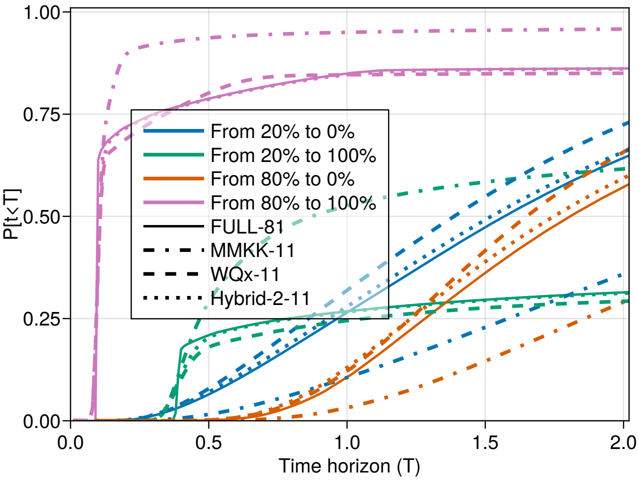


Figure 9: A look at first passage time distributions predicted by the model from a starting level to a target level. Heterogeneous rates, $L = \{81, 11\}$, $K = 5$, load=60%, and starting state $N(0) = 00000$.

trouble with the lower queue sizes in this model. WQx is much closer to the full model, though not quite as good as Hybrid-2. The ability to follow the sharp changes in the CDF will always be a compromise between the time taken to solve the model and the accuracy.

The state the system starts in significantly influences the resulting queue CDF. Fig. 7 shows the CDF when the system starts in states $N(0) = \{00000, 10000, 11000, 11100\}$, where 0 represents a path in NLoS, and 1 represents a path in LoS. Since a practical control system needs to find a solution quickly, we show the condensed models with a less precise time horizon. Note how MMKK performs much worse when the system starts at a state that is not uniquely part of its model; MMKK only exactly models state 00000 and 11111 (states 0 and K in Fig. 3), using summaries of all the states in between. Hybrid-2 is the most accurate of the approximate models while the starting number of LoS paths are not more than the number it models fully (2 in this case), overlapping the FULL line. Starting with higher numbers of LoS has Hybrid-2 overlap with WQx. Overall WQx seems to perform almost as well as the Hybrid-2, with significantly less model states. This is because the start states are of high significance when the time horizon is short, which it is for our purpose. Less accurate time horizons do not capture sharp CDF changes well, the degree depending on the starting state.

The CDF is a summary of the dynamics of the system from the starting state for the duration of the time horizon. An infinite time horizon ($T \to \infty$) depicts the steady state behaviour, while short time horizons summarise the dynamics in the immediate future. Fig. 8 illustrates this with a starting state of 11000. Since this is a stable queueing system with a load of 60%, as the time horizon is extended, the queue distribution tends toward empty. Since the starting state is not all LoS nor all NLoS, the MMKK model performs very poorly with respect to the full model. Note that the difference between 0.5 s and 1.0 s is small, indicating we are nearing the steady state solution for this scenario. This, however, will depend on the starting state; starting with all NLoS will require a longer time horizon before the queue gets close to the steady state distribution. For control purposes, shorter time horizons are more useful so long as they can be calculated fast enough (see § 4.1) and are not too short for path management control to respond.

Although the CDF over longer time periods of time is not helpful for control purposes, the probabilities of crossing particular buffer levels, i.e., First Passage Times (FPT), may be. For example, if a QoS violation occurs, it may be important to know the probability that the queue will reach a particular level within a certain time. Fig. 9 shows the probabilities of first passage times for the following from/to pairs $\{[20, 0], [20, 100], [80, 0], [80, 100]\}\%$.[7] Since the system starts with all paths in NLoS, it will initially grow, but because the system is stable with a load of 60%, the queue will drift toward empty. This limits the probability of reaching capacity as the time increases. Note that WQx-11 begins to diverge from FULL-81 as T increases—beyond real-time control intervals. MMKK diverges more quickly, but is close to FULL-81 at very small T since in this particular scenario, all paths in NLoS is not a summary state (it is state 0, see Fig. 3). Hybrid-2-11 diverges least from FULL-81 as T increases; highlighting the relative value of the FULL model when there are fewer LoS paths, and the effectiveness of summarizing the states when there are higher numbers of LoS paths.

4.1. Accuracy and performance statistics

To more thoroughly evaluate the relative accuracy of the models, we compare the mean absolute deviation (meanAD) of a 500 point CDF of FULL, $L = 81$, with less accurate time horizons and the more compact models; all for a time horizon $T = 200$ ms. The percentiles represent the results

[7] Note that 1×10^{-10} is used to represent a queue size of 0 since the model requires the target queue size to be > 0 in order to solve it.

Table 2

Mean absolute deviation to FULL ($L = 81$, $T = 0.2$)

Percentiles of the meanAD for 500 points of the CDF [0%,100%] from 500 runs. Load varies randomly between 60-90. Starting state and rates also vary randomly. All models evaluate the same random state for the different values of L before it changes for the next iteration.

K	L	Full		MMKK		WQx		Hybrid-2	
		50%	90%	50%	90%	50%	90%	50%	90%
2	81	0.0	0.0	0.012	0.093	0.0032	0.0096	–	–
	41	0.001	0.0026	0.013	0.093	0.0044	0.011	–	–
	21	0.003	0.0077	0.015	0.093	0.0069	0.014	–	–
	11	0.0069	0.017	0.023	0.093	0.011	0.022	–	–
3	81	0.0	0.0	0.045	0.13	0.0029	0.0095	–	–
	41	0.0011	0.0027	0.045	0.13	0.0043	0.011	–	–
	21	0.0032	0.0084	0.045	0.13	0.0067	0.014	–	–
	11	0.0073	0.019	0.045	0.13	0.011	0.023	–	–
4	81	0.0	0.0	0.056	0.14	0.0039	0.011	0.001	0.0066
	41	0.0011	0.0029	0.056	0.14	0.0049	0.013	0.0023	0.008
	21	0.0033	0.0085	0.056	0.14	0.0072	0.016	0.0049	0.012
	11	0.0076	0.02	0.057	0.14	0.012	0.024	0.0088	0.021
5	81	0.0	0.0	0.067	0.17	0.0049	0.012	0.0024	0.023
	41	0.0011	0.0032	0.067	0.17	0.0063	0.014	0.004	0.024
	21	0.0035	0.0098	0.067	0.17	0.0093	0.018	0.0072	0.027
	11	0.0082	0.022	0.067	0.17	0.014	0.027	0.012	0.035

from 500 runs, where each model for a given L is solved for the same random configuration. This configuration includes: (1) a system load drawn uniformly randomly from between 60–90%; (2) Λ, M, and C_{LoS} sampled randomly over the ranges given in Table 1; and (3) the starting state randomly chosen based on the probabilities of the different paths given the aforementioned parameters. Results are given for $K = [2, 3, 4, 5]$.[8]

Table 2 gives the median and 90^{th} percentiles for the meanAD of the 500 point buffer capacity CDF with respect to the Full-81 model. A Hybrid-2 is only useful for $K \geq 4$, so only these values are shown. Looking first at FULL, the error steadily increases as L decreases. This is slightly worse for $K = 5$, where the median error is 0.82% and the 90^{th} percentile 2.2% for $L = 11$. The MMKK model shows errors increasing as K increases, though not necessarily increasing much as L decreases; especially for $N > 2$. The M/M/K/K approximation is not good enough for the time horizon accuracy to be significant. This error distribution for MMKK has a very heavy tail with a 99^{th} percentile error of 26% (not in the table). Weighting the transitions and flow rates (WQ) improves on MMKK, but not greatly (not shown in the table). This is because the starting state and initial transitions are highly significant when the time horizon is relatively short (in this case 200 ms). WQx includes this information, resulting in a significant improvement for the cost of three extra states. In the worst case ($K = 5$, $L = 11$), the median error of 1.4% is not too much worse than that of FULL for the same L. The Hybrid-2 provides some improvements over WQx. For larger K, when the starting state is not in the FULL part, the benefits of Hybrid decrease.

Table 3 shows the model execution times in seconds.[9] Firstly, note that the execution times do not seem to have particularly heavy tails. This means that the model can be solved within a relatively predictable time. The state space and the execution times increase with larger K and larger L (see § 4.1). It makes sense to use FULL for $K \leq 2$ since it is the most accurate and for such a small state space the execution time is comparable to the other models. For $K > 2$, the differences between the models become apparent. MMKK (and WQ, not in the table) have the smallest state space and the fastest evaluation times, but are inaccurate. WQx ($L = 11$) has execution times within 21 ms (90^{th} percentile) for $K = 5$, and good accuracies with respect to FULL ($L = 81$), making it the best time/accuracy compromise for use in real-time control when $K > 2$.

5. Proof of concept simulation study

As a proof of concept, we use event-based simulations to test the efficacy of a predictive multipath mmWave proxy mechanism for achieving reliable consistent communication. We envisage a scenario where a mobile real-time interactive application (e.g., immersive 3D video, UHD augmented reality, etc.) needs to communicate[10] at a constant rate of 2 Gbps with a very low delay.

[8] The solving time for FULL, $K \geq 6$ is too long for evaluation.

[9] Manjaro Linux laptop, i7-8665U, Julia 1.7.2 using MKL.jl—the IntelBLAS linear algebra libraries. These times are a little lower than in [18] which used a pre-release of Julia 1.7.

[10] We only look at the receive (downlink) direction in this example.

Table 3
Single timed process runs to measure the performance for 3 CDF points, 0%,20%,100%, in seconds
As in Table 2, percentiles from 500 randomly varied runs

K	L	Full		MMKK		WQx		Hybrid-2	
		50%	90%	50%	90%	50%	90%	50%	90%
2	81	0.22	0.45	0.11	0.11	0.19	0.23	–	–
	41	0.045	0.078	0.024	0.026	0.041	0.048	–	–
	21	0.013	0.015	0.0081	0.0086	0.012	0.014	–	–
	11	0.004	0.0049	0.0027	0.0029	0.0037	0.0047	–	–
3	81	1.3	2.4	0.2	0.23	0.55	0.68	–	–
	41	0.23	0.44	0.043	0.048	0.1	0.13	–	–
	21	0.061	0.073	0.013	0.015	0.031	0.034	–	–
	11	0.017	0.018	0.0039	0.0048	0.0088	0.0098	–	–
4	81	9.5	18.0	0.35	0.39	0.94	1.2	5.0	6.2
	41	1.4	2.6	0.071	0.077	0.17	0.22	0.79	0.93
	21	0.28	0.53	0.021	0.023	0.047	0.057	0.17	0.2
	11	0.071	0.11	0.0066	0.0072	0.013	0.016	0.045	0.052
5	81	84.0	92.0	0.57	0.61	1.6	1.9	18.0	20.0
	41	10.0	12.0	0.11	0.11	0.27	0.31	2.5	2.8
	21	1.6	1.8	0.032	0.034	0.071	0.079	0.5	0.53
	11	0.31	0.33	0.0094	0.01	0.019	0.021	0.11	0.12

5.1. Simulation scenarios

The complete control system, as depicted in Fig. 1, would integrate both path management and sender rate control to maintain a reliable consistent service balancing the various costs involved. This is not a trivial enhancement, and will be addressed in future work (see § 6.3). In this simple feasibility study, we consider just the path manager, adding and removing mmWave paths to maintain a constant QoS as our testing scenario; i.e., trying to operate at the minimum number of paths that will maintain a certain QoS. We first tested four simple methods (see Table 4 for a description of parameters):

1. **Fixed paths:** In this scenario we adopt the 99th percentile of paths used in the reactive control scenario (see next method). This is the simplest control scenario since there is *no* dynamic path selection. It is assumed that an "oracle" has chosen a priori the smallest subset of paths required to sustain an *average* channel capacity higher than the target rate of 2 Gbps.

2. **Reactive control (see Alg. 1):** If an arriving packet causes the proxy queue to cross a threshold (Q_HT), add the path with the highest available capacity. If an arriving packet finds the proxy queue empty (Q_LT[11]), remove the lowest-capacity path from the set of used paths. Changes can be no faster than the Path Change Limit (PCL) and requested changes take a Path Change Delay (PCD) to come into operation.

3. **Distribution based Predictive control (see Alg. 2):** Based on the proxy queue distribution (i.e., the probability that the queue will be less than a particular threshold over the time horizon, T). Changes take PCD plus model-solving computational costs (CC) to come into operation.

4. **FPT based Predictive control (see Alg. 3):** Based on the probability of the queue crossing particular thresholds within the time horizon (i.e., the probability that the first passage time is within the time horizon, T). Changes take PCD plus CC to come into operation.

Adding a path always increases the total available capacity, improving the QoS. However, removing a path has the potential to have a negative impact. The predictive controls have the ability to check the impact of the reduced number of paths before applying the change. However, doing this requires the model to be solved twice. When there are large numbers of paths this processing cost could be significant, but at the same time when there are large numbers of paths the proportional impact of removing the lowest capacity path is less significant. For these experiments we choose a path removal check threshold for doing this extra test so that the cost of the double check is no more than the calculation cost for the maximum number of paths, 5 in this case.

Our goal in these basic algorithms is not to try to optimize a given method, but instead, using a simple instantiation of each, illustrate both the feasibility and the potential benefits that a predictive control method may offer.

5.2. Simulator characteristics

We simulate a scenario where a sender seeks to send at 2 Gbps and there are up to 8 available mmWave paths with varying capacities and NLoS↔LoS rates. Our event-based simulator models packet transmissions. We assume that adding and removing paths takes a little time (e.g., time

[11] Test upon arrival of a packet, so empty means Q_LT = 1.

Table 4
Simulation parameters

Available mmWave paths (K)	8[12]
mmWave channel bandwidth	400 MHz
Target channel rate	2 Gbps
Distance from base station	60 m[13]
Path loss model	UMi [13]
Channel update interval	50 ms
Packet size	1500 B
Predictive Control Interval (PCI)	150 ms
Path change limit (PCL)	150 ms
Path change delay (PCD)	20 ms
Predictive time horizon (T)	200 ms[14]
Computational cost estimate (CC)	90% pctl[15]
Shadow fading time	20 ms[16]
Average time in LoS	LinRange(3,4,AC) s
Average time in NLoS	LinRange(4,3,AC) s
Path removal check threshold (PRT)	5 paths[17]
Q_LT	250 or 1 packet[18]
Q_HT	500 packets
Q_T_prob	99%

[12] This number gives a good illustration of the possible dynamics and is not unreasonable in a dense UMi scenario or in what could be deployed temporarily in an emergency situation.

[13] A 3D building map, base station placement, and movement would be more realistic, however the results are then very scenario dependent. We choose to change a minimum of parameters so that it is easier to interpret the proof of concept results.

[14] Calculations and actions take some time. T should be a little longer than the PCI.

[15] Actual time costs could be used here, but that makes the results unrepeatable, depending on background computer activity. We use $\text{CC}(K) = \{1.5, 4.8, 9.6, 16, 21, 26, 33, 40\}$ ms, 90% percentile results based on a 1000 run performance (§ 4.1) for FULL $K = \{1, 2\}$ and WQx $K = \{3, 4, 5, 6, 7, 8\}$.

[16] In a real system fading effects would occur more randomly, but final capacity would be quantized depending on the particular modulation scheme used for the given SNR. This gives a simple model of the dynamic nature of the capacity.

[17] If there are not many concurrent paths, removing one path potentially removes a large proportion of the current capacity. In this case the predictive algorithms check if removing a path will lead to problems.

[18] 250 for Alg. 2, 1 for the others.

to bring interfaces up from standby and configure routing), counting it as a 20 ms delay (PCD) in the simulation. We also assume that the work involved in changing paths means that there will be an operator limit on how often this can be done. For simplicity, we make this limit the same as the predictive control interval in this simulation. The other delay cost we consider is the time taken to solve the model (CC). Although we could use actual calculation times, we instead use costs based on the 90% percentile of 1000 runs so that the simulation results are repeatable.[19] The sum of these costs delay the chosen action (add a path or remove a path).

We choose the scenario of a mobile device in a city landscape surrounded by buildings by varying the path loss according to the standard UMi - Street Canyon model described in [13]. This is a more challenging landscape for mmWave channels, but a likely scenario for dense mmWave deployment, providing a wide selection of possible paths to different base stations.

We calculate the capacity of each path using the Shannon-Hartley theorem with a base signal-to-noise ratio (SNR) discounted by the path loss model[20]. The channel model is primed by calculating the SNR that will yield a target channel rate of 2 Gbps during LoS operation at a distance of 60 m from the base station. The LoS/NLoS state of each channel is updated independently according to a two-state Markov model (see Table 4). The channel rates are recalculated for both LoS and NLoS for each available channel every 50 ms, according to the model's normally distributed shadow fading parameter.

The predictive controls use L=11, with FULL when there are two or fewer paths, and WQx when there are more paths. The model is parametrised by whatever the current LoS and NLoS path capacities are. In a real system one of these would be known, and the other would need to be estimated. We use the actual average LoS/NLoS transition rates in the simulation. A real system will need to measure these or use an estimate based on historical data.

We choose thresholds with the objective of a fair reasonable comparison rather than what may be optimal for a particular algorithm. In the full system, choice of thresholds will depend on the QoS requirements of the application, balanced with the cost particular control actions have (see § 6.3).

Algorithm 1: Traditional reactive control

```
Every Arriving packet
    if now > LastControlTime + PCL then
        if QueueSize > Q_HT then
            Add path with highest capacity
            LastControlTime = now
        else if QueueSize < Q_LT ∧ more than one
          path then
            Remove path with lowest capacity
            LastControlTime = now
```

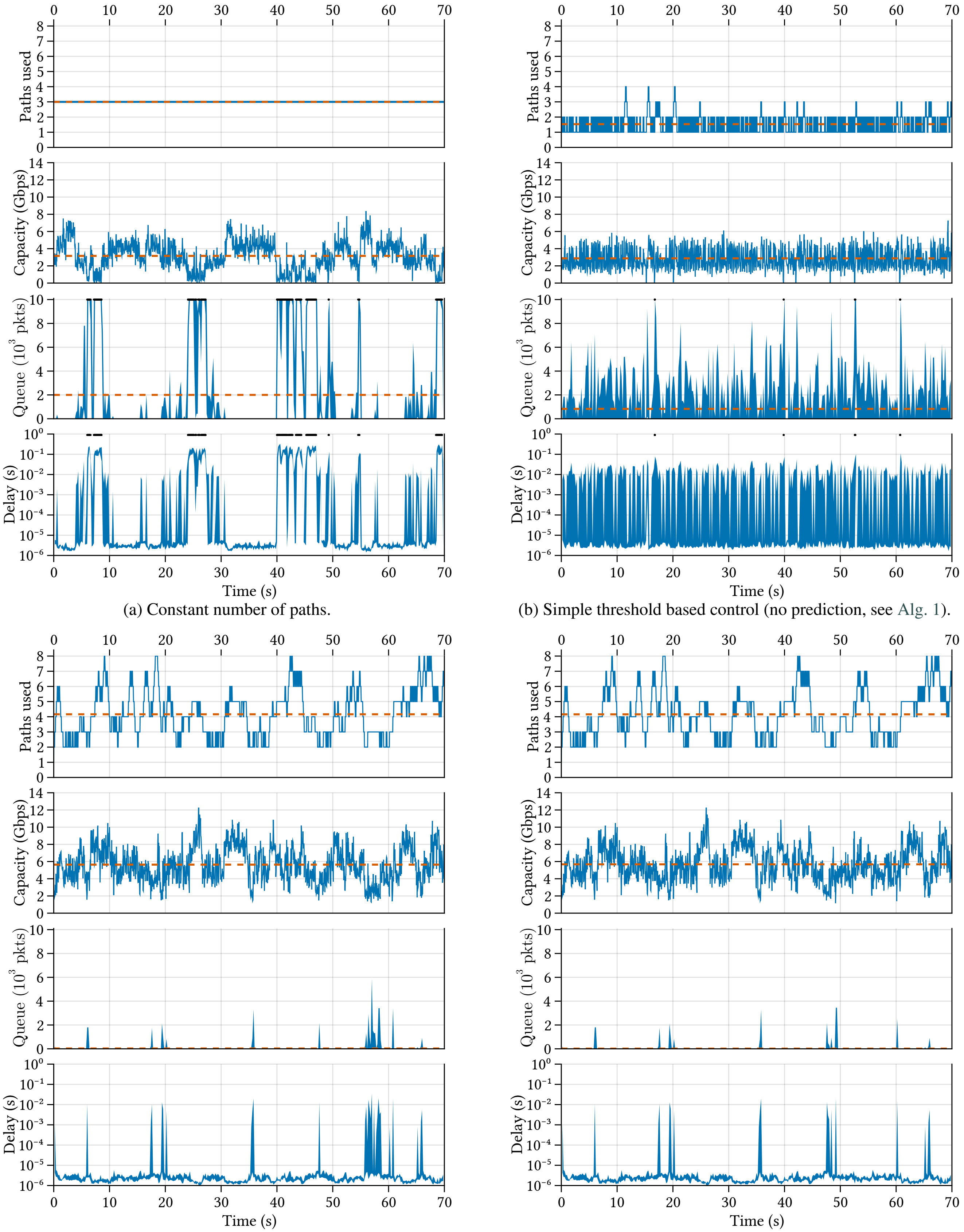
5.3. Results with the four basic methods

Fig. 10 shows the simulation results of the four methods explained in § 5.1. For each scenario we show a graph of the number of paths being used, the available capacity of the used paths, and the queue size (sampled every 200 ms, but plotted as a band of max to min achieved in that period), and the packet delay within the proxy (sampled every 200 ms, but plotted as a band of max to min achieved in that period).

In the fixed-paths scenario (Fig. 10a), we can see that the choice of 3 paths does yield an aggregate average capacity of 3.16 Gbps, way higher than the 2 Gbps sent by

[19] Hardware in a real deployment will likely be more powerful with hardware assisted matrix operations. Using laptop generated CCs provide a very conservative base.

[20] We ignore modulation changes, so capacity is continuous rather than stepped.



(a) Constant number of paths.

(b) Simple threshold based control (no prediction, see Alg. 1).

(c) CDF based predictive control (see Alg. 2).

(d) FPT based predictive control (see Alg. 3).

Figure 10: Examples of the different algorithms operations. Each scenario shows paths used, combined capacity, queue length, and delay (time in the proxy). Notes: The queue graph plots a band for the range (min,max) of queue size experienced by the system over 200 ms intervals. Time averaged queue size shown with a dashed red horizontal line. Packet loss is indicated by black markers at the queue limit (10k packets) or 1 s delay point.

Algorithm 2: Distribution based predictive control

```
Every PCI
    Update model parameters
      (i.e. queue size, current capacities and states)
    % Compute the probabilities P_L and P_H with respect to
    % the thresholds Q_LT and Q_HT, respectively, using Eq. (1)
    P_L ← Solve(model state, Q_LT)
    P_H ← Solve(model state, Q_HT)
    if P_H < Q_T_prob then
        % When the probability P_H that the queue
        % will be ≤ Q_HT after T is < Q_T_prob,
        % we need to add a path
        Add the path with the highest capacity
    else if P_L > Q_T_prob ∧ more than one path
     then
        % When the probability P_L that the queue
        % will be ≤ Q_LT after T is > Q_T_prob,
        % if feasible we remove a path
        if using ≤ PRT paths then
            % Test effect of removing lowest capacity path
            P_T ← Solve(test model state, Q_HT)
            if (P_T ≥ Q_T_prob) then
                Remove path with lowest capacity
        else
            Remove path with lowest capacity
```

Algorithm 3: First passage time based predictive control[21]

```
Every PCI
    Update model parameters
      (i.e. queue size, current capacities and states)
    if QueueSize ≥ Q_HT then
        % Calculate the probability P_T with respect to
        % the threshold Q_LT, using Eq. (2)
        P_T ← Solve(model state, Q_LT)
        if P_T < Q_T_prob then
            % The queue level is ≥ Q_HT, and the probability
            % P_T that the queue goes down to Q_LT within T
            % is < Q_T_prob, so we need to add a path
            Add the path with the highest capacity
    else
        P_T ← Solve(model state, Q_HT)
        if (1 - P_T) < Q_T_prob then
            % When probability P_T that the queue remains
            % below Q_HT during T is < Q_T_prob,
            % we need to add a path
            Add the path with the highest capacity
        else if using more than 1 path then
            if using ≤ PRT paths then
                % Test effect of removing lowest capacity path
                % to make sure it is safe to do so
                P_T ← Solve(test model state, Q_HT)
                if (1 - P_T) > Q_T_prob then
                    Remove path with lowest capacity
            else
                Remove path with lowest capacity
```

the source. However, there is a non negligible chance of one or more paths being in NLoS at some points in time, and the combined capacity being less than 2 Gbps. As a result, despite the high average capacity, congestion occurs — and, worse, bursty packet losses — since the aggregate capacity sometimes falls below the 2 Gbps target for fairly long periods. This is reflected similarly in the packet delay in the proxy.

Having a path-control policy is integral to improving the situation. Fig. 10b illustrates a simple queue threshold based scheme described by Alg. 1. This results in an average number of paths of 1.53. One path is often enough, if it is in LoS, and the resulting aggregate average capacity of 2.88 Gbps is slightly lower than in the fixed-paths case. Even though the average queue is shorter and the losses much less bursty than for the fixed-paths case, there remains extensive queueing for much of the time. The reactive control helps to mitigate the delay to some extent since there are not the high delay spikes seen in the constant path scenario. However, there is still significant delay for most of the simulation time because reactive control only happens after the delay occurs. Abrupt changes in capacity due to LoS/NLoS and shadow fading, combined with the time to effect path changes, cannot be mitigated by a purely reactive control.

The results with a predictive CDF based controller, i.e., Alg. 2, are shown in Fig. 10c. By probabilistically predicting the queue distribution over a short time in the future, rather than just reacting to it, this controller is able to keep a very short queue and avoid losses altogether. The controller adds an extra path if the queue CDF over the next 200 ms is predicted to have more than a 1% chance (i.e., 1−Q_T_prob) of being over the 500 packet threshold, and removes a path if the queue CDF has a more than 99% chance of being below 250 packets. Ensuring reliable consistent communication requires more paths. The model is not perfect, and there is that 1% chance of the queue having levels above 500 packets, but overall the predictive control maintains a reliable and consistent capacity of at least 2 Gbps for the sender. The predictive FPT based controller, i.e., Alg. 3, also manages to maintain a reliable consistent capacity (see Fig. 10d). In both of these scenarios, the queue is empty most of the time with the occasional spike in delay when the prediction is not quite right — it is probabilistic after all. The choice of which one to use in practice would depend on which best represents the particular QoS agreement/requirement of the application. For example, if QoS is best represented by the queue distribution over T, then it is better to use a CDF-based controller, whereas use of FPT is more adequate if QoS is best captured by transient threshold excursions.

Fig. 11 shows a log scale CDF of proxy delay from 100 simulations with varying random number seeds. It may seem surprising at first to see the scenario with a fixed number of paths have a higher proportion of very low delays than the no-prediction reactive mechanism. The constant 3 paths leaves the proxy queue empty for most of the time, however,

[21] The remove path element of this algorithm adopts the same check as Alg. 2, different to [18].

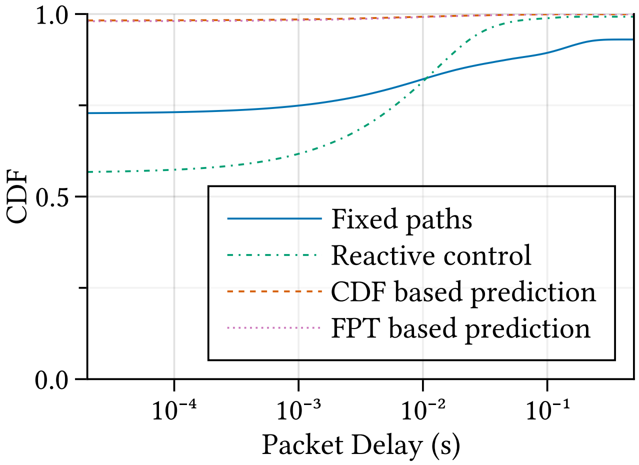


Figure 11: Delay CDF from 100 simulations. The plots for CDF-based and FPT-based prediction overlap.

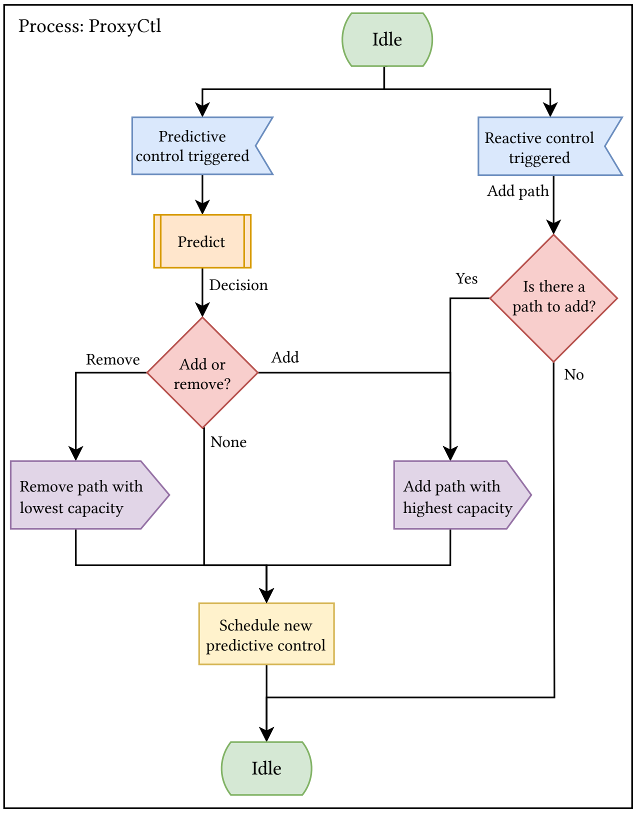

Process: ProxyCtl
Idle
Predictive control triggered
Reactive control triggered
Add path
Predict
Decision
Yes
Is there a path to add?
Remove
Add or remove?
Add
No
None
Remove path with lowest capacity
Add path with highest capacity
Schedule new predictive control
Idle

Figure 12: SDL of a prediction based control system with reactive control. In Predict, either CDF based control (see Alg. 2) or FPT based control (see Alg. 3) can be used.

when capacity drops there is no mechanism to alleviate this so queueing delay and loss becomes more significant. The no-prediction reactive mechanism does manage to reduce the packet loss and the extreme delays. The predictive controls both show very little delay since, for the most part (99%), arriving packets find the proxy queue empty.

Since the aim of this work is consistent reliable communication, it is natural to ask whether augmenting the predictive control could help to reduce the handful of delay spikes that occur. The key rationale is that if the capacity is close to correct, the queue potentially grows less quickly, making reactive control in those specific circumstances viable. We explore this proposition next.

5.4. Integrating reactive with predictive control

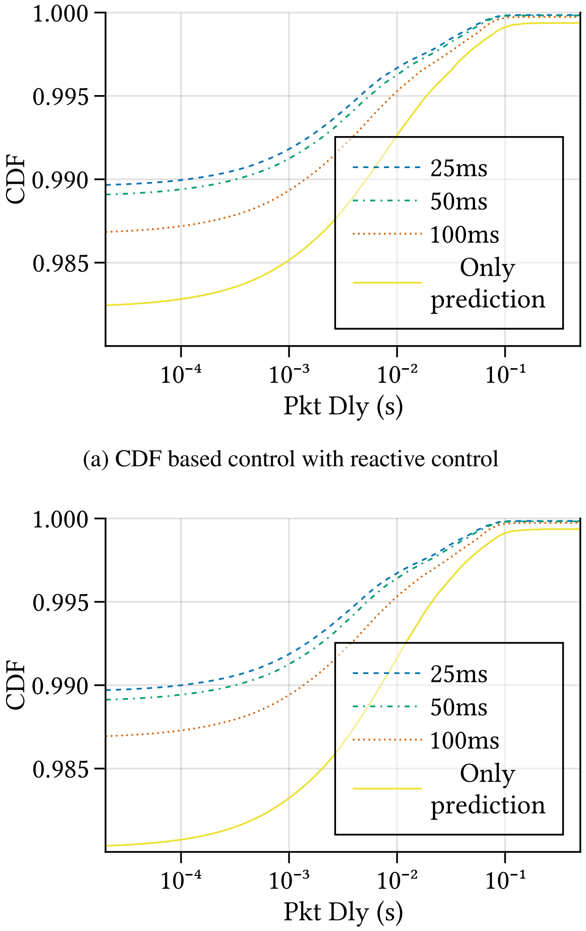
Pure prediction-based control is a good approach if the predictions made are accurate. Our predictions are probabilistic and there can also be events and system changes that are not modelled. If the predictions do not turn out to reflect what actually happens, increased queueing and higher delay can be the consequence. To mitigate the degradation of QoS this may cause, we augment the predictive-based control system with a *reactive* control element. This allows the mechanism to react to spurious spikes in the queue, by adding a path before the next scheduled predictive control, thereby helping to drain the queue.

Fig. 12 describes the revised control system, that is, prediction-based control with reactive control. In this revised system, a change in the number of available paths can be initiated by either predictive control or reactive control. The predictive control is generally triggered every 150 ms as before. The reactive control is only triggered when reality did not match the prediction, that is, when the queue is growing because we do not have enough resources and need to add a path. We check if there are paths available to add, then add the path with the highest capacity. A new predictive control is then scheduled in 150 ms. This means that every time a reactive control is triggered, the next predictive control is rescheduled some milliseconds forward. The reason we choose to shift the predictive control schedule is to avoid having a new path change by predictive control too soon after the previous reactive control path change.

To prevent the reactive control from being triggered too soon after the last path change, a limit is set for how long since the last path change we can initiate a new path change using a reactive control. This is called the *reactive control interval* T_R. This limit prevents the reactive control from being triggered too often, which may be costly for the operator and interfere with the predictive control.

We evaluated the performance of the hybrid control system with respect to T_R using the same simulation scenario as in § 5.3, measuring the delay of each packet going through the proxy. Each simulation was repeated 100 times, each run with a different seed, for each of $T_R = \{25, 50, 100\}$ ms. We do not test $T_R \leq 20$ ms since path change delay (PCD, see Table 4) is 20 ms so values below 20 ms could initiate a new path change before a previous one has been activated. We compare the results against the purely predictive algorithms described in Alg. 2 and Alg. 3.

Fig. 13a and Fig. 13b show the results when using CDF-based and FPT-based prediction, respectively. Note that we are only looking at the very top of the CDF since the delay spikes we seek to mitigate are relatively rare (see figures Fig. 10c and Fig. 10d). This also reveals a very small amount of packet loss in the prediction-only algorithms that was



(a) CDF based control with reactive control

(b) FPT based control with reactive control

Figure 13: Delay CDF from 100 simulations, using two different prediction based algorithms.

Table 5
Average number of path changes caused by a reactive control after 100 runs of the simulation. In the purely predictive control models, around 470 predictive control actions are performed in each run.

Prediction	Reactive control interval T_R		
	25 ms	50 ms	100 ms
CDF based	9.14	6.72	5.43
FPT based	8.86	6.59	5.26

not evident on the scale Fig. 11 was plotted. The impact of the reactive control interval on the higher percentiles of the delay distribution is clearly seen in Fig. 13. As T_R is reduced, the impact of the delay spikes is also reduced. The very small amount of packet loss is also reduced.

In Table 5 we can see how often on average a path was added because of a reactive control for the different values of T_R. Compared to how many times predictive control is triggered — about 470 times for the purely predictive control — we do not change the number of paths with reactive control very often. As T_R is reduced, reactive controls are triggered more often, but they are still comparatively very small. This confirms that the predictive control is enough most of the time, but also shows that the demonstrated benefit of the hybrid control comes at very little additional path change overhead.

Using this hybrid approach may also be beneficial in other scenarios. For example, if there is a way for the network to signal the proxy before a path is lost due to some network disruption, then the proxy could react to this signal and preemptively manage its paths so as to avoid any QoS impacts.

6. Discussion

6.1. Accuracy vs tractability of the models

The key to being able to use the proposed MMFQ models for predictive control, is being able to solve a chosen model quickly enough. Two factors influence this: the number of states in the Markov model and the accuracy with which the time horizon, T, is represented. We compared modelling the full system state with a number of compressed, more tractable models. A probability weighted model based on the number of paths in LoS that includes a transient fully modelled starting state and the next hop (WQx) gives highly accurate predictions with a much smaller and tractable number of states than the full model for scenarios with more than two available paths. A matrix exponential representation of T is used to represent T more compactly and accurately than with the traditional Erlang approach.

We demonstrate that a matrix exponential representation of T of order $L = 11$ is both sufficiently accurate (50th percentile of meanAD 1.2%, WQx for 4 paths with respect to the Full model with $L = 81$, see Table 2) and tractable (solved on a laptop in a median time of 13 ms for the same parameters, see Table 3). Note that in the simulation experiments in § 5 we used 90% percentiles for the computational costs (CC), rather than median costs (i.e., path addition/removal actions took even longer).

6.2. Scheduling

Our proposed mechanisms manage a number of mmWave paths to ensure there is enough capacity to carry the application traffic at its desired rate. The resulting capacity is the aggregate of any number of mmWave paths. How packets are scheduled onto the different paths impacts performance. In some sense, this scheduling problem has similarities to that commonly discussed in reference to multipath transport protocols like MPTCP and MPQuic [14, 45, 29, 20]. However there are some important differences:

1. The capacity on each of these paths changes rapidly due to fast fading, and dramatically as direct LoS is blocked and cleared. The resulting NLoS capacity can be anything from 0.1 of the LoS rate to 0.00001 of the LoS rate, or worse, depending on the blockage and particular location's topology. This leads to packet transmissions 10

to 100 000 times slower on a NLoS path than on a LoS path.

2. The wireless mmWave paths will have homogenous and very small propagation delays.

3. TCP-type congestion controls are not relevant for the applications studied here which require a consistent reliable data rate. Therefore, scheduling is making best usage of the *sufficient* capacity made available through the path management. More complex scenarios where some sort of quality compromise could be required are discussed in § 6.3.

These differences potentially limit the usefulness of current multipath scheduling techniques designed for MPTCP and MPQuic for use over less challenging networks.

For example a simple round-robin technique on two paths, one in LoS and one in NLoS, could result in 10 or more packets being sent on the LoS path to every 1 packet on the NLoS path. This causes Head of Line (HoL) blocking for ordered delivery to the application at the receiver. An out-of-order buffer at the receiver would be needed to temporarily store every packet sent over the LoS path after a packet was sent over the NLoS paths. The delay induced by this is significant, though generally not as large as queueing delays within the proxy due to lack of capacity. A scheduling algorithm that instead chooses the highest capacity first could dramatically improve on this performance, especially in circumstances where there was sufficient capacity over LoS paths to be able to mostly avoid using NLoS paths.

Despite this, path dynamics ensure that some packets will always be caught on a NLoS path. Forward Error Correction (FEC) techniques can reduce the HoL blocking problem, but at the expense of framing and encoding delays [16]. Apart from that, the very small propagation delays may make an Automatic-Repeat-Request (ARQ) technique viable on LoS paths when packets are delayed on NLoS paths, though at the expense of more complicated senders and receivers.

For reliable, consistent, high data rate, low latency, real-time applications, these delays will necessitate a receiver jitter-buffer dimensioned to handle delays to the 99th percentile or higher. This adds a fixed and possibly large delay to the path. Reducing the average delay is therefore not of great benefit; delays to 99th percentile have to be reduced to keep the jitter-buffer small. The goal in our future work here is to find the best scheduling/FEC/ARQ combination that can achieve this.

6.3. Balancing network costs and QoS

The full control system depicted in Fig. 1 has two possible actions: change the number of paths to maintain the required rate, and/or adjust the send rate and impact the application QoS. Optimizing this choice involves weighing the costs. Example costs of changing the number of paths are: the impact on power consumption of the mobile device and network costs of managing additional paths–perhaps even with other operators. Indeed, the action of adding and removing paths means bringing up network interfaces and putting them in standby, which takes time and may incur higher energy consumption. There may also be operator limits on how often paths should be changed and how many paths can be used at different times and in different places. Choosing appropriate thresholds to balance these changing costs with the application's QoS requirements, and deciding the optimal action (path change or send rate change) at a given instant is necessary for the deployed system.

The problem of balancing cost and performance can be approached from different angles and using different controller types. Since we have already obtained a predictive model of the system, one of the candidate controllers is Model Predictive Control (MPC) [8]. MPC is one of the successful controllers used for optimizing processes over time. Its main elements are: 1) a predictive model, 2) a temporal window of optimization, and 3) feedback correction. The main benefit of MPC is that it optimizes the current timeslot, while taking future timeslots into account, which has parallels with our models derived in the paper. It can also incorporate constraints in the optimization problem. It is common to utilize a closed-form model in MPC. Due to the problem complexity, we did not obtain such a model. However, our estimates that are numerically derived from the predictive model can be fed into an MPC-like controller.

The other approach to solving such a problem is defining a constrained optimization problem [7] with the goal of maximizing a utility function. The function can be defined as an increasing function of current send rate. It can also incorporate all the costs as a negative term. QoS requirements, e.g., send rate and delay, are defined as the constraints of the optimization problem. This type of controller does not necessarily need to consider a future horizon. There are concerns that should be taken into account on defining such an optimization problem on, for example, its tractability, to spend less time and energy, when the control frequency increases. Indeed, making the utility function concave or convex can be of great help since concavity (convexity) can ensure that all local optima are global optima and the problem becomes more tractable.

In this paper, we have only considered a path controller to ensure consistent communication. In the case where senders also actively change their rate via a controller (e.g., a congestion controller), how these two controllers affect each other and their stability is of concern, which should be examined.

As future work we plan to pose this balancing task for the above controllers by defining a utility function, with QoS requirements and costs as constraints to allocate paths and/or adjust send rates. We will integrate this with the MMFQ predictive control mechanism we have developed in this paper. This will perform the *action optimization* function block depicted in Fig. 1, feeding back new threshold values to the MMFQ model block and choosing the optimal action for the circumstances expected over the next control interval. In addition, mutual effects between this controller and the sender's controller need to be studied.

7. Conclusions

Reliable, consistent and very high data rate mobile communication will become especially important for services such as future emergency communication. Millimeter-wave technology provides the needed capacity, however it lacks the required reliability due to the abrupt capacity changes any one path experiences. Intelligently making use of varying numbers of available mmWave paths, perhaps even through multi-operator agreements; and balancing mobile power consumption with path costs and the need for reliable consistent quality will be critical to attaining this aim. This paper provides the first step, showing that our model-based reliability prediction is indeed useful and computationally feasible.

We model mmWave path blocking with two states, LoS and NLoS, combining these states for the available paths into a Markov model. This then drives a fluid queue to model buffer occupancy at the proxy. The transient solution to this model allows us to look at either the queue distribution over the next T s or the probability of crossing a particular buffer level within the next T s. This short term prediction allows the system to react to potential problems before they happen, thus maintaining reliable consistent communication.

Our proof of concept tests with two simple, proactive path control algorithms demonstrate the potential effectiveness of the proposed predictive approach especially compared to static or non-predictive controls. Further, we illustrate the potential benefits of a hybrid predictive/reactive path control algorithm, combining proactive and reactive control.

Our next step for the control system (see Fig. 1) is to develop the "action optimization" block. This block will use feedback to dynamically bridge the model–reality gap, and will balance the costs, choosing the best action for the given circumstances. We will then build a working control system. Further steps beyond the control system involve investigating appropriate, adaptable multipath scheduling methods and dealing with missing packets, whether due to loss or an abrupt link speed change, through erasure coding[22] (see [48]). Our long term aim is a fully functioning and deployable multipath mmWave proxy for reliable consistent communication.

Acknowledgements

The authors would like to thank Gabor Horvath for providing early access to [3] and help in understanding their Matlab code, especially with respect to the transient first passage time calculations. P. Teymoori was part-funded by the Research Council of Norway under its "Toppforsk" programme through the "OCARINA" project (https://www.mn.uio.no/ifi/english/research/projects/ocarina/). The authors would also like to thank the anonymous reviewers for their suggestions to improve the paper.

References

[1] 3GPP 2020. *23.501:System Architecture for the 5G System*. 3GPP. v16.4.

[2] S. Ahn and V. Ramaswami. 2004. Transient Analysis of Fluid Flow Models via Stochastic Coupling to a Queue. *Stochastic Models* 20, 1 (2004), 71–101.

[3] N. Akar, O. Gursoy, G. Horvath, and M. Telek. 2020. Transient and First Passage Time Distributions for First and Second-order Multi-regime Markov Fluid Queues via ME-fication. *Methodology and Computing in Applied Probability* 23 (2020), 1257–1283.

[4] N. Akar and K. Sohraby. 2004. Infinite- and finite-buffer Markov fluid queues: A unified analysis. *Journal of Applied Probability* 41 (2004), 557–569.

[5] Q. An, Y. Liu, Y. Ma, and Z. Li. 2020. *Multipath Extension for QUIC*. Internet-Draft draft-an-multipath-quic-00. IETF.

[6] Apple. 2020. Improving Network Reliability Using Multipath TCP. https://developer.apple.com/documentation/foundation/urlsessionconfiguration/improving_network_reliability_using_multipath_tcp

[7] S. Boyd and L. Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[8] E. F. Camacho and C. Bordons Alba. 2013. *Model predictive control*. Springer Science & Business Media.

[9] Q. De Coninck and O. Bonaventure. 2020. *Multipath Extensions for QUIC (MP-QUIC)*. Internet-Draft draft-deconinck-quic-multipath-06. IETF.

[10] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure. 2016. A First Analysis of Multipath TCP on Smartphones. In *Proc. of Passive and Active Measurement (PAM)*. Springer, 57–69.

[11] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure. 2016. Observing real smartphone applications over multipath TCP. *IEEE Commun. Mag.* 54, 3 (2016), 88–93.

[12] J. Deutschmann, K.-S. Hielscher, and R. German. 2020. *Multipath Communication with Satellite and Terrestrial Links*. Internet-Draft draft-deutschmann-sat-ter-multipath-00. IETF.

[13] ETSI 2020. *5G; Study on channel model for frequencies from 0.5 to 100 GHz*. ETSI. v16.1.0.

[14] Simone Ferlin, Özgü Alay, Olivier Mehani, and Roksana Boreli. 2016. BLEST: Blocking Estimation-based MPTCP Scheduler for Heterogeneous Networks. In *Proc. of IFIP Networking*. IEEE, 431–439.

[15] S. Ferlin, T. Dreibholz, and Ö. Alay. 2014. Multi-path transport over heterogeneous wireless networks: Does it really pay off?. In *Proc. of IEEE GLOBECOM*. ACM, 4807–4813.

[16] Simone Ferlin, Stepan Kucera, Holger Claussen, and Özgü Alay. 2018. MPTCP Meets FEC: Supporting Latency-Sensitive Applications Over Heterogeneous Networks. *IEEE/ACM Trans. Netw.* 26, 5 (2018), 1–14.

[17] D. A. Hayes, D. Ros, and Ö. Alay. 2019. On the importance of TCP splitting proxies for future 5G mmWave communications. In *Poc. IEEE LCN, Symposium on Emerging Topics in Networking*. IEEE, 108–116.

[18] D. A. Hayes, D. Ros, Ö. Alay, and P. Teymoori. 2021. Reliable Consistent Multipath mmWave Communication. In *Proc. of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '21)*. ACM, 149–148.

[19] G. Horváth, I. Horváth, and M. Telek. 2020. High order concentrated matrix-exponential distributions. *Stochastic Models* 36, 2 (2020), 176–192.

[20] Per Hurtig, Karl-Johan Grinnemo, Anna Brunstrom, Simone Ferlin, Özgü Alay, and Nicolas Kuhn. 2019. Low-Latency Scheduling in MPTCP. *IEEE/ACM Trans. Netw.* 27, 1 (2019), 302–315.

[21] J. Hwang and J. Yoo. 2015. Packet scheduling for multipath TCP. In *Proc. IEEE Int'l Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 177–179.

[22] N. Keukeleire, B. Hesmans, and O. Bonaventure. 2020. Increasing Broadband Reach with Hybrid Access Networks. *IEEE Commun. Stand. Mag.* 4, 1 (2020), 43–49.

[23] I. Khan, M. Ghoshal, S. Aggarwal, D. Koutsonikolas, and J. Widmer. 2022. Multipath TCP in Smartphones Equipped with Millimeter Wave Radios. In *Proc. of ACM Workshop on Wireless Network*

[22] Sometimes called network coding.

Testbeds, Experimental evaluation & CHaracterization. ACM, 54–60.

[24] M. Kim, S.-W. Ko, H. Kim, S. Kim, and S.-L.Kim. 2018. Exploiting Caching for Millimeter-Wave TCP Networks: Gain Analysis and Practical Design. *IEEE Access* 6 (2018), 69769–69781.

[25] Minho Kim, Seung-Woo Ko, and Seong-Lyun Kim. 2017. Enhancing TCP end-to-end performance in millimeter-wave communications. In *Proc. of IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 1–5.

[26] V. Kulkarni and B. Garbinato. 2019. 20 Years of Mobility Modeling & Prediction: Trends, Shortcomings & Perspectives. In *Proc. of ACM SIGSPATIAL*. ACM, 492–495.

[27] V. G. Kulkarni. 1998. *Frontiers in Queueing: Models and Applications in Science and Engineering*. CRC Press, Inc., USA, Chapter Fluid Models for Single Buffer Systems, 321—338.

[28] Y.-S. Lim, Y.-C. Chen, E.M Nahum, D. Towsley, and K.-W. Lee. 2014. Cross-layer path management in multi-path transport protocol for mobile devices. In *Proc. of the IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 1815–1823.

[29] Yeon-sup Lim, Erich M. Nahum, Don Towsley, and Richard J. Gibbens. 2017. ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths. In *Proc. of ACM CoNEXT*. ACM, 147—-159.

[30] Y. Liu, Y. Ma, C. Huitema, Q. An, and Z. Li. 2021. *Multipath Extension for QUIC*. Internet-Draft draft-liu-multipath-quic-03. IETF.

[31] G. R. MacCartney, T. S. Rappaport, and S. Rangan. 2017. Rapid Fading Due to Human Blockage in Pedestrian Crowds at 5G Millimeter-Wave Frequencies. In *Proc. of IEEE GLOBECOM*. IEEE, 1–7.

[32] Pablo Jimenez Mateo, Claudio Fiandrino, and Joerg Widmer. 2019. Analysis of TCP Performance in 5G mm-Wave Mobile Networks. In *Proc. of IEEE ICC*. IEEE, 1–7.

[33] S. Mohebi, F. Michelinakis, A. Elmokashfi, O. Grøndalen, K. Mahmood, and A. Zanella. 2021. Sectors, Beams and Environmental Impact on Commercial 5G mmWave Cell Coverage: an Empirical Study. *arXiv:2104.06188 [cs.NI]* (2021).

[34] A. Narayanan et al. 2020. A First Look at Commercial 5G Performance on Smartphones. In *Proc. of The Web Conference*. ACM, 894–905.

[35] A. Nikravesh, Y. Guo, F. Qian, Z.M. Mao, and S. Sen. 2016. An in-depth understanding of multipath TCP on mobile devices: Measurement and system design. In *Proc. of ACM MOBICOM*. ACM, 189–201.

[36] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure. 2012. Exploring mobile/WiFi handover with multipath TCP. In *Proc. of ACM SIGCOMM workshop on Cellular networks*. ACM, 31–36.

[37] M. Polese, R. Jana, and M. Zorzi. 2017. TCP in 5G mmWave Networks: Link Level Retransmissions and MP-TCP. In *Proc. of the IEEE International Conference on Computer Communications (INFOCOM), Workshops*. IEEE, 343–348.

[38] M. Polese, M. Mezzavilla, M. Zhang, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi. 2017. milliProxy: A TCP proxy architecture for 5G mmWave cellular systems. In *Proc. of Asilomar Conference on Signals, Systems, and Computers*. IEEE, 951–957.

[39] R. Poorzare and A. Calveras Augé. 2021. FB-TCP: a 5G mmWave friendly TCP for urban deployments. *IEEE Access* 9 (2021), 82812–82832.

[40] R. Poorzare and A. Calveras Augé. 2021. How sufficient is TCP when deployed in 5G mmWave networks over the urban deployment? *IEEE Access* 9 (2021), 36342–36355.

[41] V. Ramaswami, D.G. Woolford, and D.A. Stanford. 2008. The erlangization method for Markovian fluid flows. *Annals of Operations Research* 160 (2008), 215–225.

[42] Y. Ren, W. Yang, X. Zhou, H. Chen, and B. Liu. 2021. A survey on TCP over mmWave. *Computer Communications* 171 (2021), 80–88.

[43] B. Sericola. 1998. Transient analysis of stochastic fluid models. *Performance Evaluation* 32, 4 (1998), 245–263.

[44] H. Sinky, B. Hamdaoui, and M. Guizani. 2016. Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks. *IEEE Trans. Wireless Commun.* 15, 7 (2016), 4754–4764.

[45] H. Wu, Ö. Alay, A. Brunström, S. Ferlin, and G. Caso. 2020. Peekaboo: Learning-based Multipath Scheduling for Dynamic Heterogeneous Environments. *IEEE J. Sel. Areas Commun.* 38, 10 (Oct. 2020), 2295–2310.

[46] H. Wu, G. Caso, S. Ferlin, Ö. Alay, and A. Brunstrom. 2021. Multipath Scheduling for 5G Networks: Evaluation and Outlook. *IEEE Commun. Mag.* 59, 4 (2021), 44–50.

[47] H. Wu, S. Ferlin, G. Caso, Ö. Alay, and A. Brunstrom. 2021. A Survey on Multipath Transport Protocols Towards 5G Access Traffic Steering, Switching and Splitting. *IEEE Access* 9 (2021), 164417–164439.

[48] R.W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang. 2006. Network Coding Theory. *Foundations and Trend® in Communications and Information Theory* 2, 4 and 5 (2006), 241–381.

[49] Menglei Zhang, Michele Polese, Marco Mezzavilla, Jing Zhu, Sundeep Rangan, Shivendra Panwar, and Michele Zorzi. 2019. Will TCP Work in mmWave 5G Cellular Networks? *IEEE Commun. Mag.* 57, 1 (2019), 65–71.

[50] R. Zullo, A. Pescapè, K. Edeline, and B. Donnet. 2019. Hic Sunt Proxies: Unveiling Proxy Phenomena in Mobile Networks. In *Proc. of the Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 227–232.