Reliable Consistent Multipath mmWave Communication

David A. Hayes
davidh@simula.no
SimulaMet
Oslo, Norway

David Ros
dros@simula.no
Simula
Oslo, Norway

Özgü Alay
ozgua@ifi.uio.no
University of Oslo
Oslo, Norway

Peyman Teymoori
peymant@ifi.uio.no
University of Oslo
Oslo, Norway

ABSTRACT

Reliable consistent communication over millimeter-wave (mmWave) channels is a challenging problem due to their sensitivity to blocking of Line of Sight connections. MmWave is a key building block in 5G and future generation cellular networks, making solutions to this problem space important. Our aim is to use predictive control to manage and simultaneously use multiple available mmWave paths to achieve reliable consistent communication (i.e., steady transmission rate with low delay) with a multipath proxy. To this end we investigate transient solutions of Markov Modulated Fluid Queue models (MMFQ), apt because the mmWave blocking has been modeled with Markovian models. 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 of either proxy queue distributions or probabilities of reaching proxy buffer levels over a short time horizon, enabling the proxy to make preemptive path decisions to maintain a desired Quality of Service. A proof of concept simulation study demonstrates the efficacy of our proposed MMFQ-based predictive approach over either static or purely reactive control approaches.

CCS CONCEPTS

• **Networks → Middle boxes / network appliances**; **Network performance modeling**; **Network performance analysis**.

KEYWORDS

Proxy; mmWave; multipath; mobile networks; 5G

ACM Reference Format:
David A. Hayes, David Ros, Özgü Alay, and Peyman Teymoori. 2021. Reliable Consistent Multipath mmWave Communication. In *Proceedings of the 24th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '21), November 22–26, 2021, Alicante, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3479239.3485684

1 INTRODUCTION

Millimeter-wave (mmWave) radio is one of the key building blocks of 5th-Generation cellular networks (5G), and will play an increasing part in upcoming 6G networks and beyond. With 5G New Radio (NR) access technology, 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 vapor [24], and require Line-of-sight (LoS) propagation, being easily blocked by walls, foliage and people, resulting in huge fluctuations in bit rates [29]. However, upcoming use cases such as advanced emergency communications and augmented reality applications demand interactive, high-definition video communication with high bit rates that are as stable and consistent as possible. At the same time, as these applications may be latency-sensitive, they may benefit from stable low delays. Facilitating stable reliable communication for such demanding applications over mmWave links is a challenging problem; one that will escalate as coming mobile systems (6G/7G) rely more on mmWave links at even higher frequencies than today.

Performance-enhancement proxies (PEPs), and in particular TCP splitting PEPs (SPEPs), are widely deployed in cellular networks [36] 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 [13] 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.

Another approach to provide reliability and performance improvements in wireless links has been the use of multipath transport protocols. These have seen not only significant research efforts but also wide-scale, real-world deployment [5, 6]. In 5G, the proposed Access Traffic Steering, Switching, and Splitting (ATSSS) architecture [1] provides 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 [28] 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 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

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

MSWiM '21, November 22–26, 2021, Alicante, Spain

ACM ISBN 978-1-4503-9077-4/21/11…$15.00
https://doi.org/10.1145/3479239.3485684

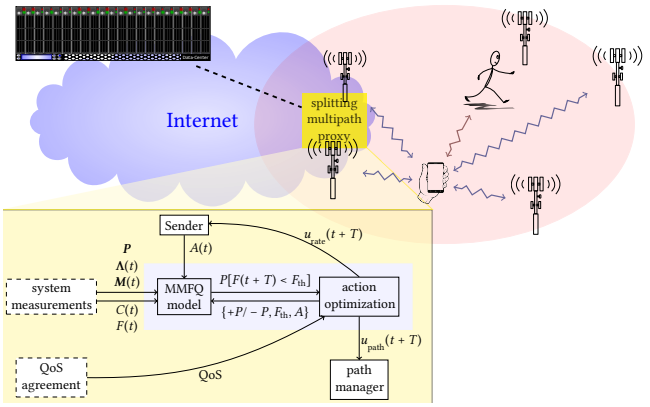


Figure 1: System architecture overview of mmWave scenario with five base stations connected to the multipath proxy. For details on the proxy control system, see Secs. 3 and 5.[2]

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 UE. We propose the use of a *splitting multipath proxy* that separates the multipath mmWave domain from the Internet with two key roles: (1) for a given application and UE, select the minimum number of mmWave links necessary to provide the required QoS, and (2) schedule data packets for transmission across the different selected links. In this paper, we will focus solely on the *multipath management* problem (1), leaving the scheduling problem (2) for future work.

Our contributions in this paper are threefold: (1) In § 3, we develop a relatively simple *mathematical model* that tries 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. Through event-based simulations, we show the effectiveness of the proposed mechanism.

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 [20], 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 [32] 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, 30]. 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 [14]. Recent work [3] has built on this to allow efficient numerical solutions, timely enough for control purposes.

mmWave Communications' Impact on Transport Protocols: [25] 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. [24] presents 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.

[31] report that TCP performance over mmWave is seriously impaired by drastic channel changes between LoS and NLoS . Several approaches have been proposed to address this, all based on some type of TCP proxy that considers the properties of mmWave channels [17, 18, 29]. 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 [8, 9, 26] have motivated multipath adoption in 5G. There are two main approaches to 5G multi-connectivity via multipath transport solutions: *Above-the-Core* and *Core-Centric* [34]. 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. 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 [10] [16], 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 approach.

Multipath Path Management: This could be 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., [27, 33]). Path termination is considered in several proposals, by taking into account e.g. in-order packet delivery [12], RTT differences between paths [15], or MAC-layer information about a link's status [21]. Or in our case using multiple paths simultaneously as part of the same connection.

Multipath transports adopt different strategies for path management. With MPQUIC [7, 22], both hosts can negotiate multipath capabilities during the handshake, to set the state of and preferences

[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 network side, as well as cost constraints on the use of multiple links.

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

3 MODELLING FOR PREDICTIVE CONTROL

The key dynamic affecting quality of service over the radio link is the mmWave LoS/NLoS blocking dynamic, since a blocked path may have a capacity of less than 1% of a direct LoS path. Markov models have been used for many years to model human movement [19], and in particular mmWave blocking [23]. We base our work on a 2-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 Quality of Service (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 $\boldsymbol{N} = \{\nu_1 \dots \nu_N\}$, where each state ν_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 ν_n. This is represented by continuous time transition matrix $\boldsymbol{Q} \in \mathbb{R}^{(N+1)\times(N+1)}$, where element Q_{ij} is the rate of moving from ν_i to ν_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 LoS: $\boldsymbol{C} = \{(C^{(1)}_{\text{NLoS}}, C^{(1)}_{\text{LoS}}), \dots, (C^{(K)}_{\text{NLoS}}, C^{(K)}_{\text{LoS}})\}$. This is represented by a diagonal fluid rate (or fluid drift) matrix $\boldsymbol{R} \in \mathbb{R}^{(N+1)\times(N+1)}$, where element R_{ii} is the net rate of fluid flow into the queue when we are in the system model-state ν_i. $R_{ii} = A - D_i$, where A is the sender's transmission rate, and $D_i = \sum C_p$ the combined capacity for every path according to their NLoS/LoS path-state represented in system-state ν_i. When $R_{ii} > 0$, fluid is filling the queue at rate R_{ii} while in system-state ν_i, and when $R_{ii} < 0$ fluid is draining from the queue at rate 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 where $F(t)$ 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 [30]. However, the resulting state space can be significantly reduced by using a matrix exponential estimate ([14], 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: $P[F(t+T) \leq x \mid F(t) = a, N(t) = \xi]$;

[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.

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

[6] https://julialang.org/. Our Julia code will be made publicly available.

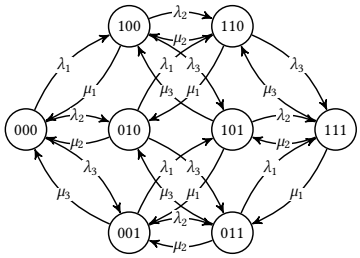


Figure 2: Full state (FULL) Markov model of LoS/NLoS state for a three base station (bs) scenario. Node labels indicate 1-LoS/0-NLoS state to each bs. λ_i is the rate of bs i changing from NLoS to LoS, and μ_i from LoS to NLoS.

(2) The probability that the first passage time is $< T$, i.e., the chance of the queue starting at a while in state ξ reaching b within T, $P[\inf_T\{F(t+T) = b \mid F(t) = a, N(t) = \xi\} < T]$.

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. 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.

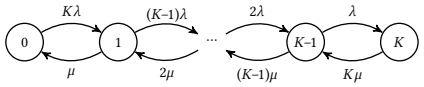


Figure 3: M/M/K/K (MMKK) model for K base stations (i.e. K paths), where each state represents the number of LoS paths.

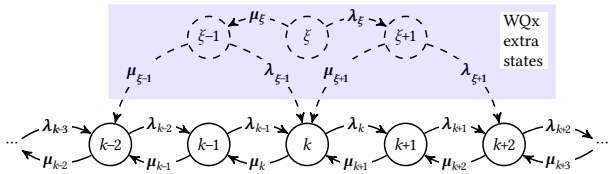


Figure 4: Weighted model (WQ) and the extended WQx model which includes an additional starting state and one jump either side (if needed).

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:

$$\boldsymbol{W}_k = \left\{ w^{(\kappa)}_{\text{LoS}} \left(1 - w^{(\kappa)}_{\text{NLoS}}\right), \kappa \in \mathcal{K}_k \right\}, \qquad \dot{\boldsymbol{W}}_k = \frac{\boldsymbol{W}_k}{\sum \boldsymbol{W}_k} \tag{1}$$

$$\boldsymbol{\Lambda}_k = \left\{ \sum \lambda_\kappa, \kappa \in \mathcal{K}_k \right\}, \qquad \boldsymbol{\lambda}_k = \sum (\dot{\boldsymbol{W}}_{k-1} \cdot \boldsymbol{\Lambda}_{k-1}) \tag{2}$$

$$\boldsymbol{M}_k = \left\{ \sum \mu_\kappa, \kappa \in \mathcal{K}_k \right\}, \qquad \boldsymbol{\mu}_k = \sum (\dot{\boldsymbol{W}}_k \cdot \boldsymbol{M}_k) \tag{3}$$

$$\mathbb{C}_k = \left\{ \left(\sum C^{(\kappa)}_{\text{LoS}}, \sum C^{(\kappa)}_{\text{NLoS}} \right), \kappa \in \mathcal{K}_k \right\}, \boldsymbol{C}_k = \sum (\dot{\boldsymbol{W}}_k \cdot \mathbb{C}_k) \tag{4}$$

where k is the number of LoS paths, $\mathcal{K}_k$ 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.

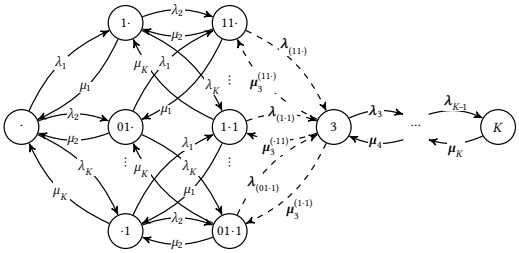


Figure 5: Example $K/2$ Hybrid FULL/WQx Markov model (Hybrid-2). System with $K > 3$ parallel paths, FULL used for states up to $J = 2$ with combinations of the remaining $K - J$ paths modelled collectively as WQ (starts in FULL).

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 the 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 2 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.

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. [14] 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. For a matrix exponential SCV < $2/L^2$ for odd L, allowing much smaller and more tractable matrices to be used.

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.

Table 1: Model characteristics for figures.

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

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

Since our goal is to have a good model for use in real-time control, the difference in accuracy between the models and the effect of L is important. 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 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. Following sharp changes in the CDF will always be a compromise between the time taken to solve the model and the accuracy.

Here we have chosen a time horizon useful for path management. Shorter time horizons may be useful for shorter time scale control, provided the models can be solved quickly enough (see § 4.1). As $T \to \infty$, the distribution moves toward the steady state distribution, i.e., the long term average behaviour, which is not useful for control. Although the CDF over longer time periods of time is not helpful for control purposes, the probabilities of crossing particular buffer levels, First Passage Times (FTP), 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. 7 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 since 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.

The state the system starts in significantly influences the resulting queue CDF. Fig. 8 shows the CDF when the system starts in

[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.

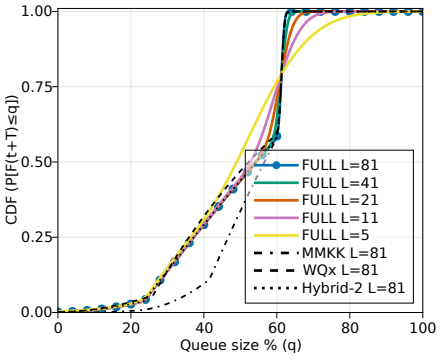


Figure 6: Relative accuracy: FULL for various T accuracies (CME levels) and condensed models. CDF over $T=0.2$ s, buffer starting at 20%, $N(0)=00000$ (all NLoS), $K=5$, and load=60%.

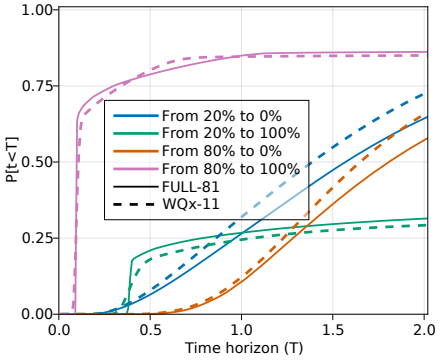


Figure 7: A look at first passage time distributions predicted by the model from a starting level to a target level. Heterogeneous rates, $L=\{81,11\}$, $K=5$, load=60%, and starting state $N(0)=00000$.

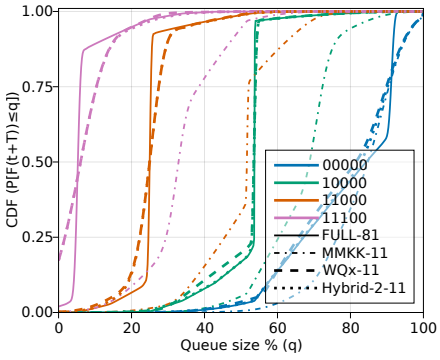


Figure 8: Impact of the starting state on the queue distribution. This is for a scenario where $K=5$, $L=\{81,11\}$, load=60%, and $T=0.2$ s.

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 summarises all states between all NLoS and all LoS). Hybrid-2 is the most accurate of the approximate models while the 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.

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 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 $L=11$. MMKK shows errors increasing as K increases, though not necessarily increasing much as L decreases; especially $N>2$. 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 the 99^{th} percentile error 26% (not in table). Weighting the transitions and flow rates (WQ), improves on MMKK, but not greatly (not shown in 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 provide some improvements over WQx. For larger K, when the starting state is not in the FULL part, the benefits of Hybrid decrease.

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

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

Table 3: Single process run for perf for 3pts (seconds)

K	L	Full		MMKK		WQx		Hybrid-2	
		50%	90%	50%	90%	50%	90%	50%	90%
2	81	0.26	0.36	0.13	0.17	0.23	0.32	–	–
	41	0.059	0.064	0.032	0.034	0.053	0.063	–	–
	21	0.019	0.02	0.011	0.012	0.015	0.019	–	–
	11	0.006	0.0072	0.0039	0.0041	0.0047	0.0069	–	–
3	81	1.4	1.8	0.24	0.33	0.65	0.81	–	–
	41	0.29	0.4	0.057	0.063	0.14	0.17	–	–
	21	0.079	0.094	0.019	0.021	0.043	0.051	–	–
	11	0.024	0.026	0.0059	0.0071	0.013	0.015	–	–
4	81	9.5	11.0	0.42	0.55	1.2	1.5	5.3	6.6
	41	1.6	1.9	0.092	0.11	0.22	0.34	1.0	1.2
	21	0.37	0.46	0.029	0.033	0.068	0.08	0.24	0.32
	11	0.092	0.11	0.0097	0.01	0.019	0.023	0.063	0.081
5	81	87.0	98.0	0.69	0.86	1.8	2.2	18.0	21.0
	41	11.0	12.0	0.14	0.16	0.36	0.43	2.8	3.3
	21	1.8	2.2	0.044	0.048	0.095	0.11	0.63	0.73
	11	0.4	0.5	0.014	0.014	0.027	0.03	0.15	0.21

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 becomes apparent. MMKK (and WQ not in table) have the smallest state space and the fastest evaluation times, but are inaccurate. WQx ($L = 11$) has execution times within 30 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.

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 § 5.4). 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 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 Q_T_prob over the time horizon). 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.

Our goal 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 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. Although we could use actual calculation times, we instead use costs based on the 90% percentile of 500 runs so that the simulation results are repeatable.[18] 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 [11]. 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[19]. The channel model is primed by calculating the

[9] Manjaro Linux laptop, i7-8665U, Julia 1.7 using the IntelBLAS linear algebra libraries.
[10] We only look at the receive (downlink) direction in this example.
[11] Test upon arrival of a packet, so empty means Q_LT = 1.
[18] Hardware in a real deployment will likely be more powerful with hardware assisted matrix operations. Using laptop generated CCs provide a very conservative base.
[19] We ignore modulation changes, so capacity is continuous rather than stepped.

Table 4: Simulation parameters

Available Channels (AC)	8
mmWave channel bandwidth	400 MHz
Target channel rate	2 Gbps
Distance from base station	60 m[12]
Path loss model	UMi [11]
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[13]
Computational cost estimate (CC)	90% pctl[14]
Shadow fading time	20 ms[15]
Average time in LoS	LinRange(3,4,AC) s
Average time in NLoS	LinRange(4,3,AC) s
Path removal check threshold	5 paths[16]
Q_LT	250 or 1 packet[17]
Q_HT (3 ms)	500 packets
Q_T_prob	99%

[12] 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.
[13] Calculations and actions take some time. T should be a little longer than the PCI.
[14] Actual time costs could be used here, but that makes the results unrepeatable, depending on background computer activity. We use $CC(K) = \{2.1, 7.7, 14, 22, 29, 38, 46, 55\}$ ms, 90% percentile results based on a 500 run performance (§ 4.1) for FULL $K = [1, 2]$ and WQx $K = [3, 8]$.
[15] 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.
[16] If there are not many concurrent paths, removing one potentially removes a large proportion of the current capacity. In this case the predictive algorithms check if removing a path will lead to problems.
[17] 250 for Alg. 2, 1 for the others.

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 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 § 5.4).

5.3 Results

Fig. 9 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).

In the fixed-paths scenario (Fig. 9a), 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 the source. However, there is a non negligible chance of one or more paths being in NLoS, 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.

Having a path-control policy is integral to improving the situation. Fig. 9b illustrates a simple queue threshold based scheme. This

Algorithm 1: Traditional reactive control

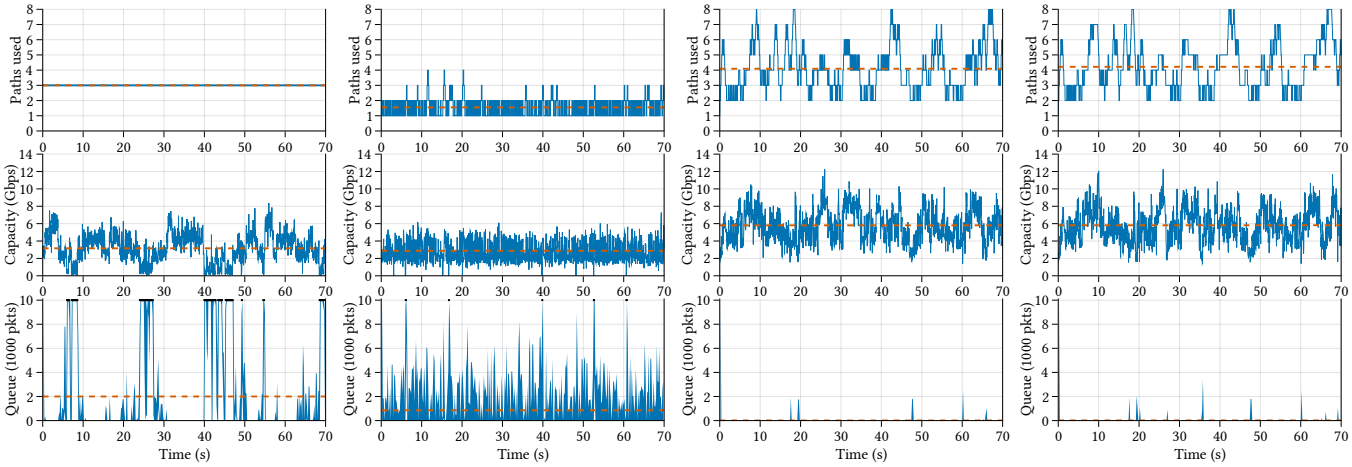
```
Every Arriving packet
    if now > LastControlTime + PCL then
        if QueueSize > Q_HT then
            Add path with highest capacity
            LastControlTime = now
        else if QueueSize < Q_LT ∧ more than one path then
            Remove path with lowest capacity
            LastControlTime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

Algorithm 2: Distribution based predictive control

```
Every PCI
    Update model parameters
     (i.e. queue size, current capacities and states)
    // Calculate the probabilities with respect to the thresholds
    (P_L, P_H) ← Solve(model state, Q_LT, Q_HT)
    if P_H < Q_T_prob then
        Add the path with the highest capacity
    else if P_L > Q_T_prob ∧ more than one path then
        if not using many paths then
            Test if removing lowest capacity path will be OK
            (P_L, P_H) ← Solve(model state, Q_LT, Q_HT) if (P_H
              ≥ Q_T_prob) then
                Remove path with lowest capacity
        if P_H < Q_T_prob then
            Remove path with lowest capacity
```

Algorithm 3: First passage time based predictive control

```
Every PCI
    Update model parameters
     (i.e. queue size, current capacities and states)
    if QueueSize ≥ Q_HT then
        // Calculate the probability with respect to the threshold
        P_T ← Solve(model state, Q_LT)
        if P_T < Q_T_prob then
            Add the path with the highest capacity
    else
        P_T ← Solve(model state, Q_HT)
        if (1 - P_T) < Q_T_prob then
            Add the path with the highest capacity
        else if more than one path then
            Test if removing lowest capacity path will be OK
            P_T ← Solve(model state, Q_HT)
            if (1 - P_T) > Q_T_prob then
                Remove path with lowest capacity
```



(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 9: Examples of the different algorithms operations.
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)

results in an average number of paths of 1.55. 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 delays for much of the time. 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 are shown in Fig. 9c. 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 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 also manages to maintain a reliable consistent capacity (see Fig. 9d). 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, is it represented best by the queue distribution over T? Use CDF. Or best by transient threshold excursions? Use FPT.

5.4 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. The action of adding and removing paths may also cost administratively, in power consumption (bringing up network interfaces and putting them in standby), and this takes time. 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 or send rate change) at a given instant is necessary for the deployed system.

As future work we plan to pose this balancing task as a utility maximization problem with QoS requirements and costs, as constraints to allocate paths and/or adjust send rates. We will integrate this with the MMFQ model-based 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.

6 CONCLUSIONS

Reliable, consistent and very high data rate mobile communication will become especially important for future services such as, among other things, future emergency communication needs. MmWave technology provides the needed capacity, however lacks the 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.

Key to being able to use this for predictive control, is being able to solve the 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 compare 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 2 available paths. A matrix exponential representation of T is used to more compactly and accurately represent T than the traditional Erlang approach.

We demonstrate that a matrix exponential representation of T of order $L = 11$ is sufficiently accurate (median 1.2%, WQx for 4 paths with respect to the Full model $L = 81$, Table 2) and tractable (solved in a median time on a laptop of 20 ms for the same parameters, Table 3). Our proof of concept tests with a simple path control algorithm demonstrate the potential effectiveness of this, especially compared to static or non-predictive controls.

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 speed change, through erasure coding[20] (see [35]). Our long term aim is a fully functioning deployable multipath mmWave proxy for reliable consistent communication.

ACKNOWLEDGMENTS

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/).

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* (2020).
[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] Q. De Coninck and O. Bonaventure. 2020. *Multipath Extensions for QUIC (MP-QUIC).* Internet-Draft draft-deconinck-quic-multipath-06. IETF.
[8] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure. 2016. A First Analysis of Multipath TCP on Smartphones. In *Passive and Active Measurement*. 57–69.
[9] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure. 2016. Observing real smartphone applications over multipath TCP. *IEEE Communications Magazine* 54, 3 (2016), 88–93.
[10] 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.
[11] ETSI 2020. *5G; Study on channel model for frequencies from 0.5 to 100 GHz.* ETSI. v16.1.0.
[12] S. Ferlin, T. Dreibholz, and Ö. Alay. 2014. Multi-path transport over heterogeneous wireless networks: Does it really pay off?. In *IEEE GLOBECOM*. 4807–4813.
[13] D. A. Hayes, D. Ros, and Ö. Alay. 2019. On the importance of TCP splitting proxies for future 5G mmWave communications. In *IEEE LCN Symposium on Emerging Topics in Networking*. 108–116.
[14] G. Horváth, I. Horváth, and M. Telek. 2020. High order concentrated matrix-exponential distributions. *Stochastic Models* 36, 2 (2020), 176–192.
[15] J. Hwang and J. Yoo. 2015. Packet scheduling for multipath TCP. In *7th Int'l Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 177–179.
[16] N. Keukeleire, B. Hesmans, and O. Bonaventure. 2020. Increasing Broadband Reach with Hybrid Access Networks. *IEEE Communications Standards Magazine* 4, 1 (2020), 43–49.
[17] 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.
[18] M. Kim, S.-W. Ko, and S.-L. Kim. 2017. Enhancing TCP End-to-End Performance in Millimeter-Wave Communications. arXiv:1709.00717 [cs.NI]
[19] V. Kulkarni and B. Garbinato. 2019. 20 Years of Mobility Modeling & Prediction: Trends, Shortcomings & Perspectives. In *Proc. of ACM SIGSPATIAL*. 492–495.
[20] 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.
[21] 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 *INFOCOM*. 1815–1823.
[22] Y. Liu, Y. Ma, C. Huitema, Q. An, and Z. Li. 2021. *Multipath Extension for QUIC.* Internet-Draft draft-liu-multipath-quic-03. IETF.
[23] 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*. 1–7.
[24] 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).
[25] A. Narayanan et al. 2020. A First Look at Commercial 5G Performance on Smartphones. In *Proc. of The Web Conference*. 894–905.
[26] 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 *Proceedings of ACM MobiCom*. ACM, 189–201.
[27] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure. 2012. Exploring mobile/WiFi handover with multipath TCP. In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks*. 31–36.
[28] M. Polese, R. Jana, and M. Zorzi. 2017. TCP in 5G mmWave Networks: Link Level Retransmissions and MP-TCP. In *IEEE INFOCOM Workshops*.
[29] 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. *51st Asilomar Conference on Signals, Systems, and Computers* (Oct 2017).
[30] 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.
[31] Y. Ren, W. Yang, X. Zhou, H. Chen, and B. Liu. 2021. A survey on TCP over mmWave. *Computer Communications* 171 (2021), 80–88.
[32] B. Sericola. 1998. Transient analysis of stochastic fluid models. *Performance Evaluation* 32, 4 (1998), 245–263.
[33] H. Sinky, B. Hamdaoui, and M. Guizani. 2016. Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks. *IEEE Transactions on Wireless Communications* 15, 7 (2016), 4754–4764.
[34] H. Wu, G. Caso, S. Ferlin, Ö. Alay, and A. Brunstrom. 2021. Multipath Scheduling for 5G Networks: Evaluation and Outlook. *IEEE Communications Magazine* 59, 4 (2021), 44–50.
[35] R.W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang. 2006. *Network Coding Theory.* Vol. 2. Now Publishers Inc., 241–381.
[36] 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)*.

[20] Sometimes called network coding.