Passive bandwidth estimation for fog computing

Kjetil M. Omang[1], Carsten Griwodz[1], and Frank T. Johnsen[2]

[1] University of Oslo (UiO), Oslo, Norway
[2] Norwegian Defence Research Establishment (FFI), Kjeller, Norway

Abstract. This paper investigates passive bandwidth estimation as a means to improve scheduling in Kubernetes-based fog computing environments. In contrast to active probing techniques, passive methods rely on observing existing traffic patterns to infer available bandwidth (ABW), offering a non-intrusive alternative suitable for dynamic and distributed systems. The study focuses on wired infrastructure networks within the cloud–fog–edge continuum.
We develop and present an approach based on a modified version of the probe gap model, evaluated in an emulated network testbed. Ten experiments were conducted under varying traffic conditions, packet loss rates, and link capacities. Timestamping artifacts and data filtering were addressed to ensure measurement integrity. The approach demonstrates reliable ABW estimation across all scenarios, confirming the viability of passive methods for bandwidth-aware scheduling in Kubernetes.

1 Introduction

Modern software is increasingly complex, often implemented as microservices and leveraging cloud-native approaches [32]. A unified approach to development, security, and operations (DevSecOps) is ideal across diverse computing environments. To manage this complexity, service orchestration tools like Kubernetes are essential, automating deployment, scaling, and resource management for containerized applications in distributed systems, particularly within microservices architectures. Cloud, fog, and edge computing form a layered approach to data processing [25]. Cloud computing provides high computational power through centralized data centers but suffers from latency, particularly for real-time or remote applications. Fog computing reduces latency by processing data closer to the user, while edge computing processes data directly at the source (e.g., sensors) to minimize delays and reduce cloud dependency.

Kubernetes (K8s), initially developed for data centers, is well-suited for cloud environments [5]. Lightweight Kubernetes distributions (e.g., K3s, MicroK8s) are being adapted for edge computing [24], though fog computing presents unique challenges [35]. We explore how passive measurement solutions can help address bandwidth estimation (BWE) in fog environments, which is a known and currently unsolved problem [7]. We consider specifically future, network-aware versions of Kubernetes schedulers [21]. This paper addresses the issue of *cross-traffic*, or *unrelated data flows competing for bandwidth*, which introduces delays due to queuing, and summarizes work from Omang's thesis [28]. We focus

on passive measurements in emulated wired networks (the fog layer) and ask: *How can data collected via passive network interface listening be used to estimate available bandwidth?*

The paper is organized as follows: Section 2 defines key terms, Section 3 reviews BWE techniques, Section 4 describes our testbed, Section 5 presents results, and Section 6 concludes the paper.

2 Understanding Bandwidth

Bandwidth refers to the maximum rate at which data can be transmitted over a network, typically measured in bits per second (bit/s). Let us define key terms **capacity** and **available bandwidth** as they relate to packet-switched networks:

Capacity defines the maximum data rate a network link or path can handle under ideal conditions. A link is the physical connection between neighboring nodes, a path is a series of links. Link capacity is defined as the maximum rate a link can carry, ignoring protocol overheads [8]. The *Nominal Physical Link Capacity* is the theoretical capacity of a link, path capacity the minimum of its constituent links' capacities. The practical effective capacity is lower due to factors like headers and error correction.

Available bandwidth (ABW) represents the unused capacity of a network path, available for additional traffic. For the entire path, the available bandwidth is the minimum ABW across all links. ABW varies dynamically based on network traffic. While ABW is often measured in an instantaneous manner, looking at it over time provides more insight. This helps to account for variations in network usage over time, making ABW a dynamic and context-dependent measure.

ABW is time-dependent and can fluctuate significantly. For example, a 1-millisecond snapshot cannot observe transport layer effects, increasing to 1-second cannot capture rapid changes in network conditions, whereas a 1-minute window might smooth out short-term congestion events. This time sensitivity must be considered in measurements to avoid misinterpretations of ABW [8].

Achievable bandwidth refers to the maximum throughput a protocol can attain over a path under current conditions. Understanding both available and achievable bandwidths is essential for accurate BWE and effective network management.

3 Bandwidth Estimation

Knowing a network's capacity and available bandwidth is helpful for network management, but directly measuring ABW is challenging. Due to random processes and the complexity of several protocol layers, it can only be estimated. BWE refers to the techniques and processes used to infer a path's capacity or available bandwidth through observations and measurements. Numerous methods have been proposed over the years, classified and compiled in comprehensive survey papers [2, 6]. Broadly, BWE methods fall into two

categories: *active* techniques and *passive* techniques. Active methods inject probe traffic into the network to measure how the network responds, whereas passive methods observe existing traffic to infer bandwidth characteristics.

3.1 Active Probing

Active probing techniques deliberately send probe packets or packet trains through the network and analyze their behavior to estimate bandwidth-related metrics. By measuring properties such as round-trip time (RTT), one-way delay (OWD), packet dispersion, or loss rate of these probes, active methods can infer the capacity or available bandwidth of a path or a specified link [2]. A wide range of BWE techniques exists, and for brevity we will not discuss these further. There are available tools that can be used to perform active probing, e.g., Iperf3, which can find achievable bandwidth on end-to-end network paths [11].

3.2 Passive Bandwidth Estimation

Passive BWE techniques are non-intrusive methods that infer network bandwidth by analyzing existing traffic rather than generating probe packets [6]. These techniques capture packet patterns, such as timings, sizes, and protocol behavior, to estimate available bandwidth and congestion state without adding load to the network. For example, TCP's adjustments to network conditions provide valuable insights, such as inter-packet gaps and congestion window size, which are analyzed to infer bandwidth [23]. Monitoring inter-arrival times and acknowledgment gaps also helps identify congestion or bottlenecks [4, 12].

Common passive BWE techniques analyze inter-packet arrival patterns, acknowledgment gaps, and throughput fluctuations to estimate residual capacity [2,3,37]. However, passive methods struggle in low-traffic scenarios, where no observable behavior exists. To overcome this, some *hybrid* techniques use minimal proactive elements to inject small amounts of traffic only when necessary, thereby remaining non-intrusive [26].

For example, a sender-side technique that uses naturally occurring packet pairs to estimate available bandwidth showed accurate results in simple networks [23]. This was enhanced with machine learning to improve accuracy in more dynamic networks [22]. Another example is SABES (Statistical Available Bandwidth Estimation), which uses statistical analysis of passive TCP measurements and neural networks to improve BWE [9].

3.3 Factors Influencing Estimation Accuracy

BWE accuracy, both active and passive, is influenced by several factors, and assumptions about traffic can lead to inaccuracies [19]. Key factors include the burstiness of cross-traffic, probe packet sizes, and measurement interval length. This section focuses on these factors in the context of passive BWE.

Available Bandwidth Variability A key challenge in BWE is that ABW is time-varying. While path capacity is fixed, ABW fluctuates as cross-traffic starts, stops, or changes rates, occurring over multiple timescales [8, 18]. Tools like Pathload [18] address this by reporting ABW as a range, with upper and lower bounds, reflecting its variability during the measurement interval [20]. This approach accounts for the oscillating nature of ABW due to network dynamics.

Cross Traffic Patterns Many estimation algorithms assume cross traffic behaves like a steady flow during the measurement interval [36], to accurately measure its impact on probe traffic. However, bursty cross traffic, with intermittent surges and large Maximum Segment Size (MSS) packets, disrupts this assumption. If not accounted for, burstiness can lead to erratic or biased results, reducing estimation accuracy [15, 29].

Packet Sizes and Probe Traffic Pattern Probe and cross-traffic characteristics play a key role in BWE accuracy, highlighting key differences between tools. Active tools control probe size and timing (e.g., packet pairs, trains, or chirps), while passive methods only observe existing traffic and infer network properties from it.

Passive tools must work with observed packet sizes, typically assuming constant sizes to link queuing delays to network capacity and cross-traffic [10,16]. Active tools, in contrast, can space probes to detect queuing onset, while passive tools analyze less predictable traffic, often requiring more data.

Passive methods can't control probe intensity, leading to varying results: measurements may be too high (e.g., iPerf), shifting ABW, or too low, lacking queuing delays. Therefore, passive tools can be as accurate as active ones when traffic is stable, but may produce invalid results under changing patterns.

Given these challenges, passive techniques often rely on statistical analysis and long observation periods, fitting models to inter-arrival times to estimate ABW. This works well for stable traffic but may struggle with dynamic or sparse conditions, resulting in wide confidence intervals or unusable estimates.

3.4 Application Environment

To address the issue of cross-traffic, or unrelated data flows competing for bandwidth we explore passive BWE for the case of traffic of distributed application with communication needs are deployed in a fog environment. Therefore, we have developed the experimental setup discussed next, which support various application demands and cross-traffic patterns.

4 Testbed and Experiment Setup

NREC (Norwegian Research and Education Cloud)[3] provided the infrastructure for our testbed, which was built using CORE (Common Open Research

[3] https://www.nrec.no/

Exp.	Cross Traffic (Size (rate))	OWD	Jitter	Bottleneck/ Upstream Capacity	Loss	Queue Size
1	8192 B (2/s)	12 ms	0 ms	5/10 Mbit/s	0 %	75
2	256 B (64/s)	12 ms	0 ms	5/10 Mbit/s	0 %	75
3	8192 B (1/s)	22 ms	2 ms	3/5 Mbit/s	2 %	30
4	128 B (64/s)	22 ms	2 ms	3/5 Mbit/s	2 %	30
5	120 B (100/s)	12 ms	0 ms	5/10 Mbit/s	0 %	75
6	120 B (20/s)	42 ms	8 ms	1/4 Mbit/s	4 %	10
7	60-180 B (10-30/s)	42 ms	8 ms	1/4 Mbit/s	4 %	10
8	512 B (16-64/s)	12 ms	0 ms	5/10 Mbit/s	0 %	75
9	8192-128 B (0.5-80/s)	12 ms	0 ms	5/10 Mbit/s	0 %	75
10	1448-128 B (2.7-80/s)	12 ms	1 ms	7/10 Mbit/s	0 %	75

Table 1. Per-experiment MGEN and CORE configuration. Each path traverses two serial bottleneck links with the same capacity; OWD and jitter are per-link, loss is random, and the queue is FIFO with the indicated size.

Emulator) [1] for network emulation and MGEN (Multi-Generator) [27] for traffic generation. This emulation provides a valuable mix of detail and repeatability.

As depicted in Figure 1, we deployed the emulation on NREC, where nodes are implemented as Linux network namespaces interconnected by emulated links. Across ten experiments we vary (i) traffic pattern (periodic+bursty, periodic+fluid, jittered/varied), (ii) per-link one-way delay and jitter, (iii) bottleneck/upstream capacities, (iv) random loss, and (v) FIFO queue size. All paths traverse two serial, equal-capacity bottlenecks, so the effective bottleneck is well defined and comparable across scenarios (Table 1). Complete details are given in the thesis [28]. The source code and experiment data is available at `https://github.com/kjetilom/network_listener`.

The necessity for real-time operation requires the use of simplified models, which may not capture the full complexity of real-world networks. These constraints must be considered when interpreting the performance of the BWE tool. They also underscore the balance between simulation accuracy and practical feasibility.

4.1 Graph Topology

We use two equal-capacity bottlenecks in series between any two communicating nodes rather than the typical single-bottleneck dumbbell. We use this topology to introduce a compounding queueing effect which can be seen in fog environments [13].

4.2 Traffic Generation and Ground Truth Measurements

Each node generated short TCP bursts separated by exponentially distributed idle intervals with an average duration of 15 seconds, over periodic UDP

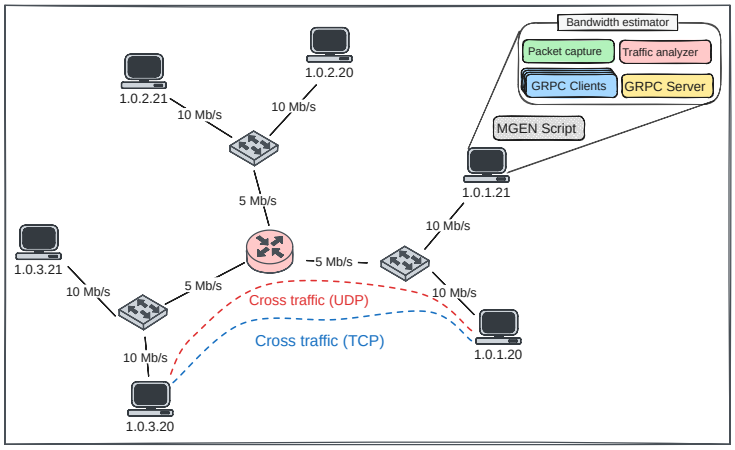

Fig. 1. Emulated topology: every communicating pair traverses two equal-capacity bottlenecks; each node runs MGEN and the estimator.

background traffic, yielding sparse cross-traffic with brief high-load episodes. To ensure each estimate spans at least one burst plus adjacent idle time, we used a 35-second measurement window for every ABW estimate.

We obtained ground truth using CORE's gRPC [14] throughput listener, sampling per-link throughput every 3 seconds. For each window, we subtracted the measured throughput from the configured capacity of each bottleneck. The path-level ground-truth ABW is the per-path minimum of these residuals.

4.3 Timestamping Artifacts and Data Filtering

Under high load we observed occasional inter-arrival gaps that implied instantaneous throughput above the physical link rate, which is physically implausible. We attribute these outliers to virtual NIC timestamping that occasionally collapses closely spaced packet arrivals. To preserve measurement integrity we apply a conservative pre-processing filter: any packet-pair gap that would imply a throughput exceeding the known link rate is discarded before regression. This removes the virtualization artifact without biasing against valid samples near the bottleneck.

5 Results

Within each 35-second window we form packet-pair gaps from observed traffic and fit a simple probe-gap relation using robust linear regression (Huber weighting) [17, 33]. The estimator generally showed a positive error relative to the true bottleneck capacity, while underestimation primarily appeared when loss or reduced TCP sending rates limited the number of usable packet-pair

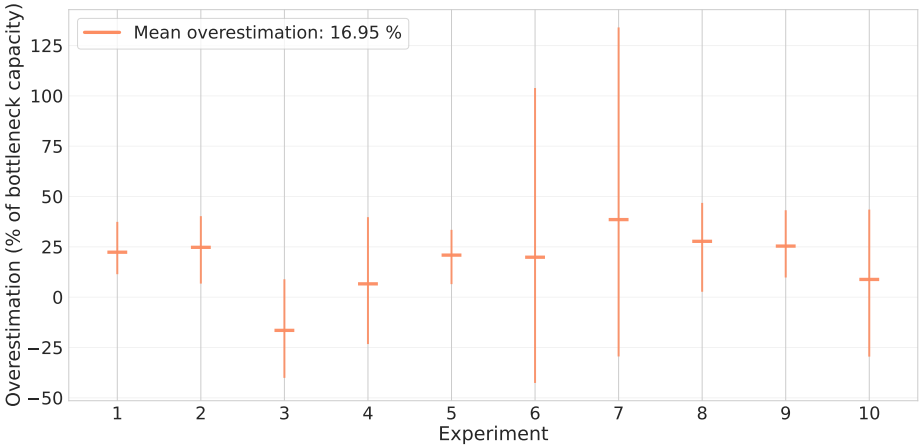

Fig. 2. Accuracy of ABW estimator by experiment: mean and 5-/95-percentiles.

samples. Figure 2 summarizes per-scenario errors as means with 95% intervals. The overall mean error was 16.95%, which is comparable with existing work [34].

5.1 Systematic Overestimation Bias

The dominant trend is a systematic overestimation of available bandwidth (typically on the order of +20-30% of bottleneck capacity). This bias stems from the combination of sparse, bursty TCP traffic and the ground-truth definition: the estimator captures only the transient high-rate intervals within bursts, while concurrent bursts on other flows crossing the same bottleneck remain unaccounted for in the momentary samples. In the baseline configuration (exp. **1**), for example, the reported ABW was ca. 22% above the ground truth.

5.2 Impact of Traffic Patterns and Network Impairments

While overestimation is the norm, its magnitude depends on traffic dynamics and impairments. In experiment **3**, configured with 2% random loss and otherwise high TCP intensity, the median error shifted to -17.8%. A similar pattern emerged when we deliberately lowered the TCP transmission rate, relative to the bottleneck capacity. In both cases, fewer packet pairs exceed the ABW, starving the estimator of the samples it needs to infer the correct gap-response relationship and lowering the estimated ABW.

In less ideal scenarios with low bottleneck capacity and a high random loss percentage, such as Experiments **6** and **7**, we saw a large increase in variability and mean error. This increase in variability is expected, as the number of measurements available for each measurement interval are significantly lowered when the capacity is lower, increasing the significance of outliers.

6 Conclusion

In this paper, we set out to explore passive available BWE in the fog (within the edge-fog-cloud continuum), guided by the idea that the natural flow of data can indicate the state of network links under changing conditions. We presented the research question: *How can data collected via passive network interface listening be used to estimate available bandwidth?* We explored this question in our literature studies and through the experimental prototyping approach presented in this paper.

Our evaluation in Section 5 demonstrated that with sufficient data and time, our tool successfully estimated ABW in most scenarios. This confirms that passive network interface listening can effectively estimate ABW. By implementing a known filtration mechanism [23], we achieved continuous, non-intrusive ABW estimation without active probing. Notably, the passive estimates rarely exceeded the true bottleneck capacity, showing the method's reliability.

We have studied passive measurements in emulated wired fog networks. Wireless networks are left for future work since they require a different methodology that accounts for environmental factors such as interference and spectrum allocation [30,31,37,38]. Emulation was chosen instead of simulation or a live fog/Kubernetes deployment, and validating the estimator in a real cluster remains work for the future.

Disclosure of Interests. The authors confirm the use of AI-powered systems to assist in the refinement of human-authored text, which is permitted under the guidelines of the call for papers. A written UiO-compliant AI usage statement is provided in [28].

References

1. Ahrenholz, J., Goff, T., Adamson, B.: Integration of the CORE and EMANE network emulators. In: Proc. of MILCOM. pp. 1870–1875 (2011). https://doi.org/10.1109/MILCOM.2011.6127585
2. Airon, M., Gupta, N.: Bandwidth Estimation Tools and Techniques: A Review (Oct 2017). https://doi.org/10.20944/preprints201710.0060.v1
3. Arsan, T.: Review of bandwidth estimation tools and application to bandwidth adaptive video streaming. In: High Capacity Optical Networks and Emerging/Enabling Technologies. pp. 152–156 (2012). https://doi.org/10.1109/HONET.2012.6421453
4. Brakmo, L., Peterson, L.: TCP Vegas: end to end congestion avoidance on a global internet. IEEE JSAC **13**(8), 1465–1480 (1995). https://doi.org/10.1109/49.464716
5. Burns, B., Grant, B., Oppenheimer, D., Brewer, E., Wilkes, J.: Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade. ACM Queue **14**(1), 70–93 (Jan 2016). https://doi.org/10.1145/2898442.2898444
6. Chaudhari, S.S., Biradar, R.C.: Survey of bandwidth estimation techniques in communication networks. Wireless Personal Communications **83**(2), 1425–1476 (2015). https://doi.org/10.1007/s11277-015-2459-2

7. Chia Chuan, W., Ul Arfeen Laghari, S., Manickam, S., Ashraf, E., Karuppayah, S.: Challenges and opportunities in fog computing scheduling: A literature review. IEEE Access **13**, 14702–14726 (2025). https://doi.org/10.1109/ACCESS.2024.3525261
8. Chimento, P., Ishac, J.: Defining Network Capacity. Request for Comments RFC5136, Internet Engineering Task Force (Feb 2008). https://doi.org/10.17487/rfc5136
9. Ciaccia, F., Romero, I., Arcas-Abella, O., Montero, D., Serral-Gracià, R., Nemirovsky, M.: SABES: Statistical Available Bandwidth EStimation from passive TCP measurements. In: Proc. of IFIP Networking. pp. 743–748 (2020)
10. Dovrolis, C., Ramanathan, P., Moore, D.: Packet-dispersion techniques and a capacity-estimation methodology. IEEE/ACM ToN **12**(6), 963–977 (2004). https://doi.org/10.1109/TNET.2004.838606
11. ESnet: iperf3 (2024), `https://software.es.net/iperf/`, release date: 13 December 2024
12. Gerla, M., Sanadidi, M., Wang, R., Zanella, A., Casetti, C., Mascolo, S.: TCP Westwood: congestion window control using bandwidth estimation. In: Proc. of GLOBECOM. vol. 3, pp. 1698–1702 (2001). https://doi.org/10.1109/GLOCOM.2001.965869
13. Goswami, A., Modi, K., Patel, C.: Evaluation of optimization algorithm for application placement problem in fog computing: A systematic review. Archives of Computational Methods in Engineering **32**(5), 2887–2915 (Jun 2025). https://doi.org/10.1007/s11831-025-10227-6, `https://doi.org/10.1007/s11831-025-10227-6`
14. gRPC Authors: Client-streaming rpc (2025), `https://grpc.io/docs/what-is-grpc/core-concepts/#client-streaming-rpc`, accessed: 2025-05-11
15. Guerrero, C.D., Labrador, M.A.: A Hidden Markov Model approach to available bandwidth estimation and monitoring. In: Proc. of IEEE Internet Network Management Workshop. pp. 1–6 (2008). https://doi.org/10.1109/INETMW.2008.4660326
16. Huang, J., Ito, N., Oshiba, T., Satoda, K., Murase, T.: PathRefiner: Accurate Bandwidth Estimation Using Large-Sized Virtual Packets for High-Speed Networks. In: Proc. of IEEE ICCE. pp. 1–6 (2024). https://doi.org/10.1109/ICCE59016.2024.10444457
17. Huber, P.J.: Robust estimation of a location parameter. The Annals of Mathematical Statistics **35**(1), 73–101 (Mar 1964). https://doi.org/10.1214/aoms/1177703732
18. Jain, M., Dovrolis, C.: End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput. ACM SIGCOMM CCR **32**(4), 295–308 (2002). https://doi.org/10.1145/964725.633054
19. Jain, M., Dovrolis, C.: Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In: Proc. of ACM IMC. pp. 272–277 (2004). https://doi.org/10.1145/1028788.1028825
20. Jain, M., Dovrolis, C.: End-to-end estimation of the available bandwidth variation range. In: Proc. of ACM SIGMETRICS. pp. 265–276 (2005). https://doi.org/10.1145/1064212.1064242
21. Keefer, L.T.: Enhancing Kubernetes Scheduling in Highly Dynamic and Unpredictable Network Environments. Master's thesis, University of Oslo, Norway (May 2025)
22. Khangura, S.K.: Machine learning-based available bandwidth estimation. Ph.D. thesis, Universität Hannover (2019). https://doi.org/10.15488/9166

23. Khangura, S.K., Fidler, M.: Available bandwidth estimation from passive TCP measurements using the probe gap model. In: Proc. of IFIP Networking and Workshops. pp. 1–9 (2017). https://doi.org/10.23919/IFIPNetworking.2017.8264826
24. Koziolek, H., Eskandani, N.: Lightweight Kubernetes distributions: A performance comparison of Microk8s, k3s, k0s, and Microshift. In: Proc. ACM/SPEC ICPE. pp. 17–29 (2023). https://doi.org/10.1145/3578244.3583737
25. Mahmud, R., Kotagiri, R., Buyya, R.: Fog Computing: A Taxonomy, Survey and Future Directions, pp. 103–130. Springer (2018). https://doi.org/10.1007/978-981-10-5861-5_5
26. Nam, S.Y., Kim, S.J., Lee, S., Kim, H.S.: Estimation of the available bandwidth ratio of a remote link or path segments. Computer Networks **57**(1), 61–77 (2013). https://doi.org/10.1016/j.comnet.2012.08.015
27. Naval Research Labratory: Multi-Generator (Aug 2021), `https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/MGEN/`
28. Omang, K.M.: Kubernetes in Edge Systems: Low Overhead Network Capacity Estimation. Master's thesis, University of Oslo, Norway (May 2025)
29. Park, K.J., Lim, H., Hou, J.C., Choi, C.H.: Feedback-assisted robust estimation of available bandwidth. Computer Networks **53**(7), 896–912 (2009). https://doi.org/10.1016/j.comnet.2008.10.023
30. Peng Zhao, Xinyu Yang, Chiyong Dong, Shusen Yang, Bhattarai, S., Wei Yu: On an efficient estimation of available bandwidth for IEEE 802.11-based wireless networks. In: Proc. of GLOBECOM. pp. 1–5 (2011). https://doi.org/10.1109/glocom.2011.6134282
31. Pogel, T., Lubbe, J., Wolf, L.: Passive client-based bandwidth and latency measurements in cellular networks. In: Proc. of INFOCOM Workshops. pp. 37–42. Orlando, FL, USA (2012). https://doi.org/10.1109/INFCOMW.2012.6193516
32. Pourmajidi, W., Zhang, L., Steinbacher, J., Erwin, T., Miranskyy, A.: A reference architecture for governance of cloud native applications. IEEE Transactions on Cloud Computing **13**(3), 935–952 (2025). https://doi.org/10.1109/TCC.2025.3578557
33. Seheult, A., Green, P., Rousseeuw, P., Leroy, A.: Robust regression and outlier detection. Journal of the Royal Statistical Society. Series A (Statistics in Society) **152**, 133 (01 1989). https://doi.org/10.2307/2982847
34. Sousa, D., Sargento, S., Luís, M.: Passive estimation of available bandwidth in heterogeneous ad hoc networks. In: 2025 IEEE 26th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). pp. 259–265 (2025). https://doi.org/10.1109/WoWMoM65615.2025.00052
35. Sri, R.L., Vetriveeran, D.: Kubernetes for fog computing - limitations and research scope. Springer Lecture notes in networks and systems **419** (2022). https://doi.org/10.1007/978-3-030-96299-9_34
36. Strauss, J., Katabi, D., Kaashoek, F.: A measurement study of available bandwidth estimation tools. In: Proc. of IMC. p. 39 (2003). https://doi.org/10.1145/948205.948211
37. Tursunova, S., Inoyatov, K., Kim, Y.T.: Cognitive passive estimation of available bandwidth (cPEAB) in overlapped IEEE 802.11 WiFi WLANs. In: Proc. of IEEE NOMS. pp. 448–454 (2010). https://doi.org/10.1109/NOMS.2010.5488496
38. Zhao, P., Yang, X., Yu, W., Dong, C., Yang, S., Bhattarai, S.: Toward efficient estimation of available bandwidth for IEEE 802.11-based wireless networks. J. Netw. and Comput. Appl. **40**, 116–125 (2014). https://doi.org/10.1016/j.jnca.2013.08.005