Dyn-WNTR: Dynamic Network Adaptive Extension for Hydraulic Simulations with WNTR

Pierluigi Locatelli, Tiziana Cattai, Simone Palumbo, Francesca Cuomo
Sapienza University of Rome, Department of Information Engineering, Electronics and Telecommunications, Italy
CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Parma, Italy
Email: {pierluigi.locatelli, tiziana.cattai, francesca.cuomo}@uniroma1.it, palumbo.1938214@studenti.uniroma1.it

***Abstract*—The Water Network Tool for Resilience (WNTR) is a Python package widely used for the simulation and analysis of Water Distribution Networks (WDNs), providing tools such as network modification, pressure-dependent demand simulation, and resilience evaluation. Although it produces time-varying data reflecting WDN activity, it requires the complete parameter configuration at the beginning of the simulation and it requires a full restart when modifying network parameters such as pipe leaks or demand variations. Here, we present Dyn-WNTR, an extension that enables dynamic adaptation of network parameters during simulation. By integrating novel functions for real-time network updates, Dyn-WNTR allows modifications, such as the introduction of leaks or demand variations, without requiring a simulation restart. This significantly improves the flexibility of hydraulic simulations, allowing more efficient scenario testing, real-time optimization, and adaptive control strategies. We also release a dataset with dynamic simulations under diverse operating conditions and network events. To encourage a broad use and future developments, the source code of Dyn-WNTR and the dataset are publicly available. By reducing computational costs and improving flexibility, Dyn-WNTR extends the capabilities of WNTR towards more advanced and real-time frameworks, particularly suited for reinforcement learning applications and digital twin modeling, where continuous interaction with the simulation environment is essential.**

***Index Terms*—water network, Internet of Things (IoT), simulation, real-time**

I. Introduction

A Water Distribution Network (WDN) is composed of interconnected elements that allow the delivery of water to end users, including domestic, industrial, and agricultural consumers. Monitoring these systems increasingly relies on Internet of Things (IoT) technologies, which enable the real-time acquisition and analysis of key physical parameters such as pressure, flow, and quality indicators [1], [2]. However, in practical scenarios, continuously monitoring the entire network is infeasible. Due to limitations in sensor availability, high deployment costs, and technical constraints, obtaining continuous and high-quality real-world data across the entire network is currently unfeasible [3]. Indeed, it is rarely possible to place sensors at all critical locations, and active components like pumps and valves introduce further complexity into the system's behavior [4], [5]. For this reason, realistic simulation tools play a crucial role in enabling the analysis and understanding of WDN dynamics under a wide range of conditions [6], [7].

Among these tools, EPANET is a widely adopted software for modeling water distribution systems [8]. Originally developed to analyze how water and dissolved elements evolve within WDNs, EPANET has become a standard for hydraulic simulations. To further extend its modeling capabilities, several open-source libraries have been developed, including WNTR (Water Network Tool for Resilience), a Python package designed for simulating and analyzing the behaviors of WDNs [9]. It provides functionalities to build WDN models and simulate network events such as leaks or demand variations, and evaluate network resilience, with pressure-dependent and demand-driven analysis. Despite the premises, traditional tools such as WNTR operate through predefined scenarios and lack the ability to interact with the simulation in real time. This static approach limits the exploration of adaptive responses or dynamic system behavior.

To address this limitation, we introduce Dyn-WNTR, an extension of the WNTR framework that brings real-time, interactive simulation capabilities to WDN modeling. Unlike already available tools, Dyn-WNTR allows users to apply changes to the network, such as modifying valve positions, altering demand patterns, or simulating failures—during the simulation itself. This interactive workflow enables immediate observation of the effects of these changes, supports adaptive strategy testing, and facilitates decision-making under dynamic conditions. By integrating this functionality directly within WNTR, Dyn-WNTR enhances the flexibility of water infrastructure analysis and significantly expands the scope of WNTR modeling.

In this work, we evaluate Dyn-WNTR on different water networks of increasing complexity. For each network, we simulate multiple demand variation scenarios, leveraging the interactive capabilities of our tool to observe the system's response in real time. Additionally, we generate dedicated datasets for each network, supporting benchmarking, machine learning applications, and further analysis. All materials, including code and datasets, are publicly available at [10] to encourage reuse within the research community.

Dyn-WNTR is particularly suitable for the development of

This work was supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP B53C22004050001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

ISBN 978-3-903176-72-0 © 2025 IFIP

integrated platforms [11] and digital twins, which are virtual representations of physical systems that evolve over time by incorporating real-time measurements [12]. In this context, the ability to simulate observed changes in the real network, such as demand fluctuations or the detection of anomalies, is fundamental. Dyn-WNTR enables researchers and water utilities to test corrective actions, such as valve adjustments or pressure regulation, directly within the simulation environment. This allows for the evaluation of intervention strategies in response to events such as leakages, to support adaptive decision-making on WDNs.

II. Background on available simulators

A variety of platforms are available for simulating water distribution systems, including both commercial and open-source solutions with overlapping but distinct features. WaterCAD [13], integrates hydraulic analysis and capacity planning into a commercial package, providing comprehensive tools, technical support, and a well-established user community. Similarly, InfoWater [14], combines simulation capabilities with GIS integration, allowing users to visualize network data in a spatial context and make decisions based on geographically linked attributes. Both are robust, industry-recognized platforms suited for organizations with resources to invest in proprietary software and the training required to master specialized interfaces. Their closed-source nature, however, and relatively high licensing costs make them unsuitable for research labs, as there is limited/abstract freedom to analyze or modify the underlying code.

By contrast, EPANET [15], developed by the U.S. Environmental Protection Agency (EPA), is a free and open-source option that remains popular for its intuitive interface, strong hydraulic and water-quality modeling capabilities, and large global user base. EPANET's longevity means there is extensive documentation and community-driven support, although the software's core functionality has changed little in recent years. Although EPANET's core functionality and user interface have remained relatively stable, its open-source nature has fostered a global community (notably through Open Water Analytics [16]) that periodically updates and extends its capabilities.

Building on EPANET's foundations is WNTR, also released by the EPA. WNTR is free and open-source but goes further by providing Python libraries designed for advanced modeling scenarios, including the analysis of network resilience under disruptive events. By integrating performance metrics for disruptive events, e.g., power outages, pipe breaks, firefighters extra demand or even earthquakes, and supporting programmable workflows, WNTR capabilities fits perfectly both to practitioner and researchers who need a modular, programmable framework to test a plethora of different scenarios. The core elements of both platforms revolve around nodes (junctions, tanks and reservoirs) and links (pipes, pumps or valves), which can be configured to simulate flow distribution, pressure variations, and water quality. While EPANET provides a user-friendly graphical environment for standard hydraulic analyses, WNTR employs Python libraries for advanced automation and resilience-oriented studies, offering a modern approach that aligns with emerging methodologies in data science and computational modeling.

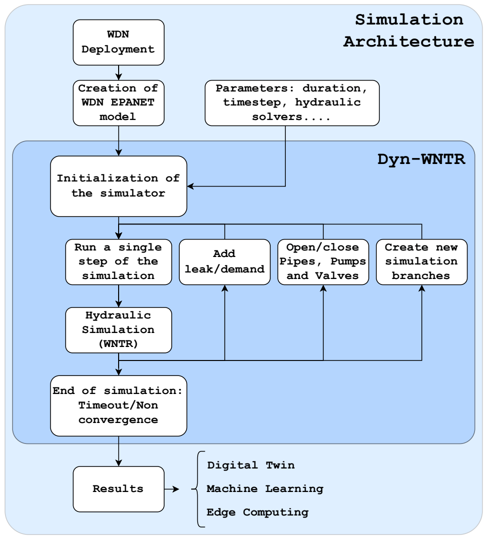

Fig. 1: Simulation architecture for WDN deployment, incorporating Dyn-WNTR simulation framework.

Additionally, other open-source projects exist in this domain, such as the Storm Water Management Model (SWMM) for combined stormwater and wastewater systems, as well as community-driven forks and enhancements of EPANET under the Open Water Analytics initiative, which further illustrate the expanding ecosystem of free modeling tools. However, given the research-oriented focus of this work, and the need for a Python-based, easily customizable toolkit, WNTR stands out as the most suitable choice, balancing practical utility with openness and extensibility.

III. Dynamic WNTR

We introduce Dyn-WNTR, an extension of WNTR to enable real-time, interactive simulations of WDNs. Unlike the traditional approach of running fixed, predefined scenarios, Dyn-WNTR enables users to introduce network changes, such as adjusting valve settings, modifying demand patterns, or simulating pipe failures, while the simulation is actively running. This dynamic interaction allows for on-the-fly exploration of system behavior and immediate observation of outcomes, making it possible to evaluate adaptive strategies and support real-time decision-making. By integrating these capabilities directly into the WNTR framework, Dyn-WNTR expands the scope of resilience modeling and brings high flexibility to water infrastructure analysis. The general architecture is reported in Figure 1 and its implementation can be found at [10].

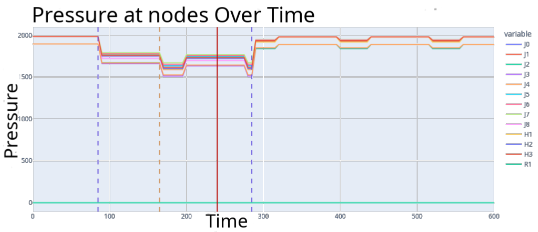

Fig. 2: Evolution of pressures in the original (main) simulation. The first two dashed lines mark the introduction of additional demand and the onset of a leak, respectively. At the solid line, the simulation timeline continues uninterrupted; the leak is repaired shortly afterward, while the added demand remains active.

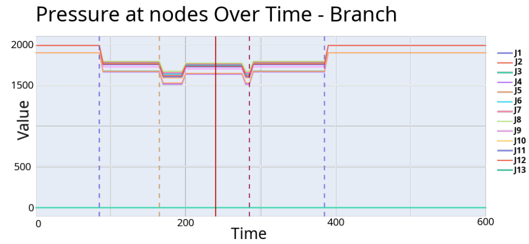

Fig. 3: Pressures in the branched simulation. The same two events occur before the solid line, but once the branch is created, the extra demand is halted immediately, and the leak is fixed only at a later stage (indicated by the final dashed line), producing different transient dynamics than in the main simulation.

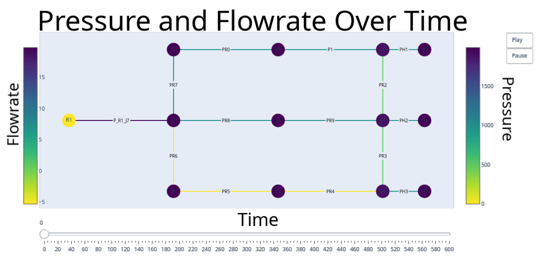

Fig. 4: Visualization of pressure and flowrate over time for the same network. Node colors reflect pressure, and pipe colors represent flowrate. The interactive controls (play, pause, and slider) allow users to explore the system's dynamics under changing demands, events, or interventions.

A. Motivation

Unlike EPANET, which supports dynamic interaction through rule-based controls and real-time inputs, WNTR requires that all simulation parameters be defined before execution. Any event, like valve actuation, pump failure, demand shift, or leak, must be pre-scheduled, preventing meaningful runtime interaction or intermediate data access. This static approach limits WNTR's effectiveness for real-time analysis, scenario testing, and Digital Twin integration.

To address this, we developed Dyn-WNTR, a dynamic extension of WNTR that enables real-time control, feedback, and state updates throughout simulation. Dyn-WNTR transforms WNTR from an offline simulator into an interactive, incremental engine capable of supporting predictive analytics, anomaly detection, and adaptive control. It allows WNTR to function not just as a planning tool, but as a responsive model that can reflect and anticipate changes in real-world water networks.

Motivated by the growing demand for Digital Twin technologies in critical infrastructure, we restructured WNTR to be interactive and time-steppable without altering its simulation core. This enables continuous integration of real-time data from physical systems, keeping the simulation aligned with the network's actual behavior. As a result, Dyn-WNTR supports accurate forecasting, responsive control, and seamless updates—without restarting the simulation at each change—making it a valuable asset for real-time decision-making and system monitoring.

B. Dynamic Enhancement

Dyn-WNTR adopts a dynamic paradigm, allowing active engagement with the simulation as it unfolds. By offering the ability to inject network events, adjust parameters, and branch scenarios without restarting the entire model, Dyn-WNTR makes it possible to evaluate how a water system might respond to rapidly changing conditions in near real time.

1) Initialization and Stepping Through the Simulation: At its core, Dyn-WNTR leverages the same hydraulic engine and data structures used in standard WNTR, ensuring consistency with prior modeling practices. When a simulation begins, Dyn-WNTR initializes an internal hydraulic model just as WNTR does, but it keeps that model in a state that can be incrementally advanced through time. Instead of running the complete simulation at once, users can "step" the model in discrete intervals, pause at intermediate states, and inspect all node and link-level properties (e.g., pressure, flow, head). This design offers fine-grained control over the simulation timeline: an operator can run the model for a short interval, induce a fault in a specific pipe, and then observe the immediate hydraulic response in the next simulation timestep.

2) Interactive Capabilities: The real distinction of Dyn-WNTR emerges in its interactive commands, which can be issued at any point during the simulation's timeline. These commands transform the model on-the-fly, without halting or re-initializing it.

a) Opening and Closing Links.: Standard WNTR allows pipe or valve statuses to be changed only by reconfiguring inputs before a run. In Dyn-WNTR, users can instantaneously open or close pipes, pumps, and valves with a single function call. The changes take effect at the next simulation step, enabling realistic simulation of repairs, maintenance, or emergency shut-offs.

b) Demand Adjustments and Spikes.: To capture realistic usage patterns or crisis conditions, Dyn-WNTR provides functions to add or remove demands at specific nodes in real time. For instance, a sudden spike in demand can be introduced to mimic a major consumer coming online, or a demand can be reduced to mirror water conservation measures. These changes feed directly into the model's hydraulic solver, granting an up-to-date view of network pressures, flows, and reservoir/tank levels.

c) Pump Outages and Valve Modifications.: Pumps often represent critical points of failure, and Dyn-WNTR offers commands to simulate pump outages. Analysts can terminate or restart pumping operations mid-simulation, then watch how water levels in downstream tanks evolve in response. Similarly, valve settings—whether pressure regulating or flow control—can be tuned on-the-fly. This flexibility suits emergency planning scenarios, where operators need to evaluate system behavior under unplanned but plausible events.

3) Branching and Parallel Scenario Exploration: A key innovation in Dyn-WNTR is its branching mechanism, which allows the current simulation state to be cloned for parallel exploration of different trajectories. Instead of rerunning the simulation from scratch, users can pause at critical moments, such as after a major leak is discovered, and create multiple branches with distinct intervention strategies. Each branch is then stepped forward in parallel, reducing simulation overhead. Figures 2 and 3 illustrate how branching enables the exploration of different strategies from a shared initial state. In both plots, the simulation progresses until the vertical solid line, where two events occur: at the first dashed line, additional demand is introduced; at the second, a leak is triggered. After the solid line, the simulation splits into two paths. In the *original* branch (top), the leak is repaired, and the additional demand continues, stabilizing the pressures. In the *branched* simulation (bottom), the demand is stopped, and the leak repair is delayed, causing more variable pressures until the leak is fixed. This approach allows for efficient exploration of what-if scenarios without recalculating the entire simulation.

4) Enhanced Visualization and Plotting: Dyn-WNTR enhances WNTR's plotting capabilities with interactive time-stepped displays. Time-series plots track variables like demand, pressure, and flowrate across nodes and links, with vertical markers for key events such as leaks and pump shutdowns (Figures 2 and 3). Additionally, an interactive network map color-codes nodes and links by metrics like pressure or flowrate, as shown in Figure 4, while still allowing the user to hover on nodes or links to extract every other useful statistic, with a slider for step-by-step exploration. These features provide both a high-level overview and detailed insights, ideal for real-time feedback, training, and demonstrations.

5) Extended Results and Analyses: Dyn-WNTR captures intermediate states at each timestep and logs user actions, allowing for detailed tracking of system behavior after events like leaks or outages. It introduces *expected_demand* and *satisfied_demand* metrics to assess supply shortfalls in real-time. These enhancements facilitate integration with machine learning models for system optimization, offering valuable insights into network performance and reliability.

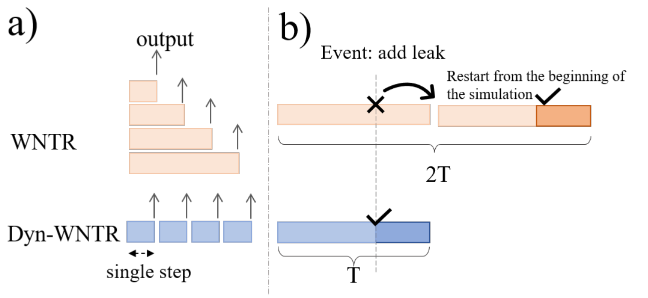

Fig. 5: Comparison between standard WNTR and Dyn-WNTR. (a) In WNTR, accessing intermediate states requires multiple simulations, which need to start over every single time; Dyn-WNTR allows direct access at each step within a single run. (b) In WNTR, adding new events requires restarting the entire simulation, while Dyn-WNTR enables event injection without restart.

C. Comparison with Standard WNTR

In many respects, Dyn-WNTR retains the key strengths of standard WNTR, notably the established EPANET-based hydraulic solver and the open-source Python environment. However, it introduces a dynamic simulation paradigm that enables adaptive control over the network state and supports interactive, step-by-step execution. This approach allows for real-time interventions and branching simulations without restarting the entire process—features that are not natively supported in standard WNTR.

Figure 5 highlights the key operational differences between Dyn-WNTR and standard WNTR. In panel (a), we consider the case in which the goal is to retrieve the system state at the end of each time step of fixed duration T_{TS}, over a total horizon divided into N_{TS} steps. In standard WNTR, this requires launching N_{TS} separate simulations, each starting from time zero and ending at time $i \cdot T_{TS}$, where i is the index of the time step ($1 \leq i \leq N_{TS}$). The cumulative simulation time required is $\sum_{i=1}^{N_{TS}} i \cdot T_{TS} = T_{TS} \cdot \frac{N_{TS}(N_{TS}+1)}{2}$. This quadratic cost in the number of steps becomes increasingly inefficient for fine-grained simulations. In contrast, Dyn-WNTR provides direct access to the state at each time step within a single run of total duration $T = N_{TS} \cdot T_{TS}$, reducing the computational cost to $\mathcal{O}(T)$.

Panel (b) illustrates another practical advantage of Dyn-WNTR when introducing a number N_E of events into the simulation at different time points. In standard WNTR, inserting N_E (e.g., leaks) at different times necessitates restarting the simulation from scratch for each scenario. This leads to a total runtime of $N_E \cdot T$. Conversely, Dyn-WNTR allows the dynamic injection of events during runtime without restarting, keeping the simulation cost constant with respect to the number of events.

On the other hand, due to the core logic of WNTR, not everything has been ported to a real-time fashion yet. Complete control over pumps and valves is still a work in progress, as the current results proved to be inconsistent or wrong, so the only way to control these nodes in the network is still to rely on the pre-programmed controls. Moreover, unlike EPANET, Dyn-WNTR does not have a proper GUI yet, which we are still developing.

IV. Dataset Generation for Comprehensive Scenario Analysis

Many water distribution studies focus on a narrow set of conditions, limiting insights into how systems behave under diverse or extreme scenarios. To address this gap, we created a new dataset comprising hundreds of Dyn-WNTR simulations designed to cover a wide range of operational states and unexpected events. The description of the steps performed to derive the datasets are reported in Algorithm 1. Our primary goal is to facilitate broad-based performance evaluation, resilience testing, and scenario planning exercises.

A. Randomized Simulation Setup

Each simulation is initialized with a standard network configuration, represented in Figure 6, but the parameters governing system behavior vary stochastically to maximize heterogeneity, as described in the dataset generation pseudocode in Algorithm 1. Specifically, we randomize:

- *Junction Demands:* Demand patterns changes so that reflect both typical daily fluctuations and abrupt surges (e.g., a large consumer coming online or spikes in usages) as shown in Figure 7.
- *Pump Working Status:* By simulating partial or full pump outages, we can capture scenarios ranging from minor inefficiencies to critical shortfalls.
- *Link Status Changes:* Pipes and valves get closed or reopened throughout the simulation to replicate field operations, maintenance activities, and sudden failures.
- *Leaks at Junctions:* Random leaks are introduced at selected junctions during the simulation, varying in both intensity and duration. These events simulate pipe bursts or small undetected leaks near consumer nodes and help evaluate how localized losses affect network-wide pressure, demand satisfaction, and recovery dynamics, as shown in Figure 8.
- *Tank Head and Operations:* Tank volumes are initialized at various states of fullness and can be adjusted over time to model interventions such as emergency refills or unplanned drawdowns.

B. Ensuring Heterogeneity

To promote a robust exploration of network responses, all parameter variations occur at random intervals or intensities rather than following fixed schedules. Some simulations introduce faults gradually—such as a creeping demand spike—while others feature abrupt events like a leak failure in the middle of a peak-demand period. This randomized approach ensures that the dataset includes both mild and severe disruptions, providing a more comprehensive view of how the water network responds to stresses.

Algorithm 1 Water Network Simulation with Random Events

```
Initialize:
D: Simulation Duration
T: Simulation Timestep
M: Water Model
N: Number of successful simulations
for each iteration i from 1 to N do
    Load: M
    Add: Simulation patterns to the model
    Initialize: Configure Simulator with D and T
    while simulation is not terminated do
        Generate random event:
        if random event is leak then
            Start or stop leak at random node
        else if random event is demand_change then
            Add or remove demand at random node
        else if random event is pipe_status_change then
            Close or open random pipe
        end if
        Advance one step in the simulation
    end while
    if simulation completed full duration then
        Save results
    else
        Report Early termination
    end if
end for
```

C. Capturing and Organizing Outputs

Each simulation run logs time-stepped data for key variables (e.g., pressure, flowrate, demand, satisfied demand) and critical events (such as pump status changes or link closures). Together with the intervention history tracked in Dyn-WNTR, these logs allow future analysts to pinpoint exactly when, where, and how conditions changed and, most importantly, how the system adapted in response. The resulting dataset is organized into scenario files, each capturing a complete simulation timeline (including intermediate states) and a metadata summary describing which random perturbations occurred.

D. Dataset Applications

Our primary motivation is to provide researchers, operators, and planners with a rich testbed for benchmarking operational strategies, testing optimization algorithms, and studying multi-event resilience. For instance, teams can replay individual scenarios to evaluate response protocols, or aggregate multiple runs to assess average performance under repeated stress conditions. By sampling from the dataset's varied and unpredictable events, water managers and policy-makers can gain

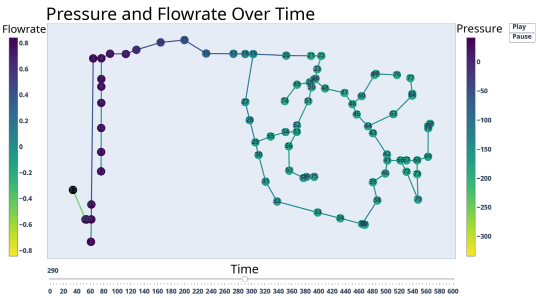

Fig. 6: Snapshot of pressure (colors on nodes) and flowrate (colors on edges) in a large WDN.

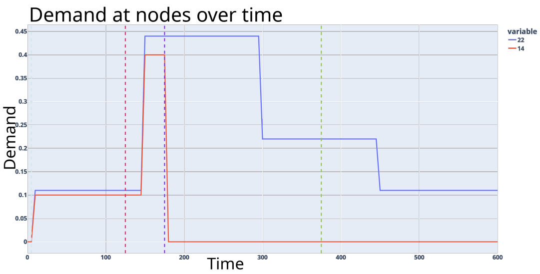

Fig. 7: The plot shows the demand evolution over time for nodes 22 and 14. Initially, both nodes follow a similar demand pattern. At the second vertical dotted line, the demand for node 14 (red line) is removed at runtime and drops to zero, while node 22 (blue line) continues its demand pattern.

deeper insights into the network's vulnerabilities, helping them prioritize investments, refine emergency response plans, and design contingency strategies that are robust under a wide array of real-world conditions.

V. Conclusions

We introduced Dyn-WNTR, a dynamic extension of WNTR that enables real-time modifications to WDN simulations, such as valve operations, demand changes, and leak insertions. This transforms WNTR's static environment into an adaptive, interactive modeling tool.

Key contributions include a software architecture for live simulation, parallel scenario branching, and enhanced real-time visualization. Compared to standard WNTR, Dyn-WNTR improves computational efficiency, supports dynamic analyses, and facilitates real-time decision-making. We also provide a dataset of randomized WDN events for benchmarking and machine learning applications. Designed for digital twin integration, Dyn-WNTR bridges live data with simulation. Future work includes full real-time node/link support and a user-friendly GUI.

References

[1] H. M. Ramos, M. C. Morani, A. Carravetta, O. Fecarrotta, K. Adeyeye, P. A. López-Jiménez, and M. Pérez-Sánchez, "New challenges towards smart systems' efficiency by digital twin in water distribution networks," *Water*, vol. 14, no. 8, p. 1304, 2022.

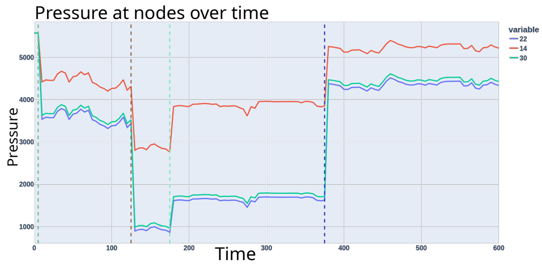

Fig. 8: In this plot, the head of the reservoir was increased to overstress the effects of the leak at node 16. Nodes 22 and 30, downstream of the leak, show significant pressure drops, while node 14, upstream, is less affected. The second dotted line indicates the removal of demand at node 14, which reduces the overall flow and lessens the water loss due to the leak. The final dotted line marks the closure of the leak, after which the pressure at all nodes begins to stabilize.

[2] A. Pagano, D. Garlisi, I. Tinnirello, F. Giuliano, G. Garbo, M. Falco, and F. Cuomo, "A survey on massive iot for water distribution systems: Challenges, simulation tools, and guidelines for large-scale deployment," *Ad Hoc Networks*, vol. 168, p. 103714, 2025.

[3] Z. Liu and Y. Kleiner, "State of the art review of inspection technologies for condition assessment of water pipes," *Measurement*, vol. 46, no. 1, pp. 1–15, 2013.

[4] T. Cattai, S. Colonnese, D. Garlisi, A. Pagano, and F. Cuomo, "Graphsmart: a method for green and accurate iot water monitoring," *ACM Transactions on Sensor Networks*, vol. 20, no. 6, pp. 1–32, 2024.

[5] G. Ö. Gararsson, F. Boem, and L. Toni, "Graph-based learning for leak detection and localisation in water distribution networks," *IFAC-PapersOnLine*, vol. 55, no. 6, pp. 661–666, 2022.

[6] A. Kaghazchi, S. M. H. Shahdany, and A. Roozbahani, "Simulation and evaluation of agricultural water distribution and delivery systems with a hybrid bayesian network model," *Agricultural Water Management*, vol. 245, p. 106578, 2021.

[7] A. Ganjidoost, M. A. Knight, A. J. Unger, and C. T. Haas, "Performance modeling and simulation for water distribution networks," *Frontiers in Water*, vol. 3, p. 718215, 2021.

[8] L. A. Rossman *et al.*, "Epanet users manual," 1994.

[9] K. A. Klise, D. Hart, D. M. Moriarty, M. L. Bynum, R. Murray, J. Burkhardt, and T. Haxton, "Water network tool for resilience (wntr) user manual," tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.

[10] Dyn-WNTRL github repository: https://github.com/rastafaninplakeibol/Dyn-WNTR.

[11] A. Pagano, D. Garlisi, F. Giuliano, T. Cattai, and F. C. Sapienza, "Swi-feed: Smart water iot framework for evaluation of energy and data in massive scenarios," in *2024 IFIP Networking Conference (IFIP Networking)*, pp. 583–585, IEEE, 2024.

[12] S. Cavalieri and S. Gambadoro, "Digital twin of a water supply system using the asset administration shell," *Sensors*, vol. 24, no. 5, p. 1360, 2024.

[13] I. Bentley Systems, "WaterCAD." https://www.bentley.com/en/products/brands/watercad. Accessed: 2025-03-28.

[14] Innovyze, "InfoWater." https://www.innovyze.com/en-us/products/infowater. Accessed: 2025-03-28.

[15] L. A. Rossman, "EPANET 2 Users Manual," tech. rep., U.S. Environmental Protection Agency, Cincinnati, OH, USA, 2000.

[16] O. W. Analytics, "Open Water Analytics Projects." https://github.com/OpenWaterAnalytics. Accessed: 2025-03-28.