CASE STUDY

How the Mississippi Department of Transportation streamlines statewide traffic video with Wowza

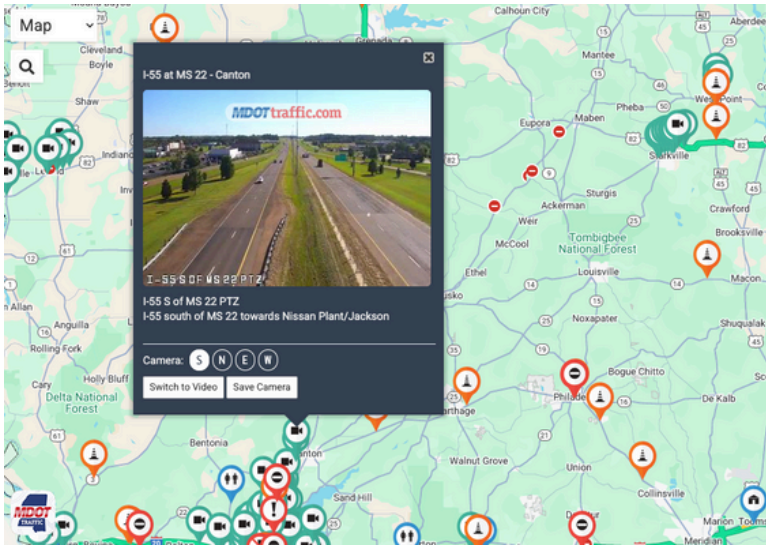
CASE STUDY: MDOT

How the Mississippi Department of Transportation Streamlines Statewide Traffic Video with Wowza Streaming Engine

1,150+ TRAFFIC CAMERAS ORCHESTRATED STATEWIDE; FEEDS AUTO-TESTED EVERY TWO HOURS.

MDOT operates a statewide traffic-camera network to keep roads safe, inform travelers, and coordinate responders. Video is mission-critical, powering incident detection, operator workflows, and public information.

Built on Wowza Streaming Engine, MDOT's system ingests IP camera feeds, automates configuration, and delivers streams to operators, first responders, and the public reliably, at scale, with caching to minimize internal bandwidth usage.



PERFORMANCE & SCALE

Ubuntu + AMD EPYC hosts tested to hundreds of streams per server, while keeping typical loads <100 per node.

OPERATIONAL AUTOMATION

Programmatically provisions and rebuilds streams via Wowza Streaming Engine APIs.

EDGE EFFICIENCY

Perimeter caching (since 2019) reduces duplicate backbone traffic and improves circuit utilization.

GOALS

- Centralize camera inventory, health checks, and control in one pane.
- Minimize manual server touches with API-driven workflows.
- Ensure resilient regional operations and smooth failover.
- Publish secure internal/high-res and public-friendly streams.

RESULTS

- Central "system of record" tightly integrated with Wowza
- Bulk camera moves/rebuilds in clicks; fewer 4 a.m. incidents.
- Predictable capacity at ~100 cameras per server (≈200 streams with two bitrates)
- Reliable dual-rendition delivery for TOC walls and public site.

THE STORY

MDOT's mission demands clear, dependable visibility across Mississippi's roads, and that vision comes to life because Wowza Streaming Engine is the operable core of its statewide video system. Today, MDOT publishes approximately 1,200 camera feeds for the public, newsrooms, and first responders, with Wowza acting as the reliable origin that powers those experiences.

Instead of hand-tuning origin nodes, MDOT built a Camera Manager that treats Wowza as a programmable fabric. Add a camera, choose a regional Wowza host, and the system provisions the application automatically. When something drifts, operators press "rebuild" to re-provision streams through Wowza, avoiding late-night server surgery. This workflow turns Wowza servers into true appliances. New capacity is as simple as running a script, registering the server, and migrating cameras in bulk, which keeps teams focused on operations rather than plumbing. If a node fails, feeds move to another Wowza host and rebuild with a button, maintaining continuity when it matters most.

Scale and performance are built in. MDOT standardizes on Ubuntu with modern AMD EPYC processors and validated high stream counts in testing. In production, the team targets about 100 cameras per origin, or roughly 200 streams, because each camera contributes two live bitrates. That headroom keeps operations predictable, even when Wowza Transcoder is generating still images for maps and listings alongside live streaming. The shift to AMD EPYC in recent years improved per node efficiency under MDOT's workload, especially with concurrent still generation.

Placement also matters. Wowza origin clusters are distributed across four regions (North, Central, South, and Hattiesburg), so MDOT can assign each camera to a geographically close server to shorten the path and reduce unnecessary backbone traffic. Choosing the right origin per device becomes an operational decision managed centrally in the interface, not a rebuild project.

On delivery, Wowza remains the authoritative origin while MDOT's NGINX perimeter caches serve the public site efficiently. A first viewer may see a brief delay on a cache miss, but subsequent viewers are served locally, which prevents multiple identical flows from traversing state networks and keeps Wowza nodes focused on origin duties. Since introducing perimeter caching in 2019, MDOT has seen significant improvements in circuit utilization.

Together, these choices, including automation through Wowza's configuration model, pragmatic scaling on proven hardware, region-aware origin placement, cache friendly delivery, and integrated still generation, translate into what matters for MDOT: dependable visibility delivered efficiently and a team freed to focus on incident response and traveler information rather than maintaining streaming infrastructure. Wowza Streaming Engine makes that operating model possible and repeatable.

> WITH WOWZA, SERVERS BECAME APPLIANCES. WE SCRIPT, PROVISION, MOVE, AND REBUILD CAMERAS THROUGH THE API, KEEP OUR VIDEO WALLS FED, AND **LET OPERATORS FOCUS ON INCIDENTS INSTEAD OF PLUMBING.**"
>
> **CLINTJOHNSON, MDOT**

FIND OUT HOW OTHERS ARE UNLOCKING THE VALUE OF VIDEO

WOWZA.COM