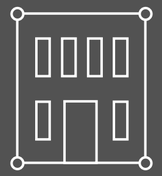
MODΣ

THE IDEAL NETWORK AT **EVERY MOMENT**™

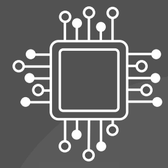
Any App

Any Cloud



Any Location

Any Distance



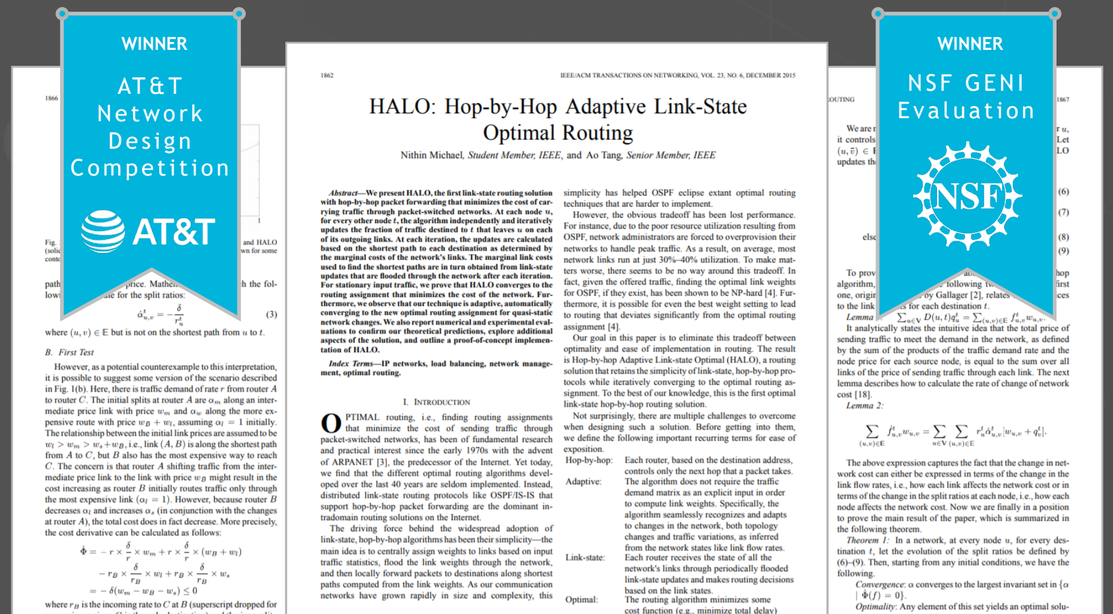
Any Device

Crafted from MODΣ AUTONOMOUS FABRIC

Millisecond Control: Key to Optimal Efficiency

Eliminates sub-second network problems

Mode created the world's only mathematically optimal, distributed feedback control algorithm.



1862 IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 23, NO. 6, DECEMBER 2015

HALO: Hop-by-Hop Adaptive Link-State Optimal Routing

Nithin Michael, *Student Member, IEEE*, and Ao Tang, *Senior Member, IEEE*

***Abstract*—We present HALO, the first link-state routing solution with hop-by-hop packet forwarding that minimizes the cost of carrying traffic through packet-switched networks. At each node u, for every other node t, the algorithm independently and iteratively updates the fraction of traffic destined to t that leaves u on each of its outgoing links. At each iteration, the updates are calculated based on the shortest path to each destination as determined by the marginal costs of the network's links. The marginal link costs used to find the shortest paths are in turn obtained from link-state updates that are flooded through the network after each iteration. For stationary input traffic, we prove that HALO converges to the routing assignment that minimizes the cost of the network. Furthermore, we observe that our technique is adaptive, automatically converging to the new optimal routing assignment for quasi-static network changes. We also report numerical and experimental evaluations to confirm our theoretical predictions, explore additional aspects of the solution, and outline a proof-of-concept implementation of HALO.**

***Index Terms*—IP networks, load balancing, network management, optimal routing.**

I. Introduction

OPTIMAL routing, i.e., finding routing assignments that minimize the cost of sending traffic through packet-switched networks, has been of fundamental research and practical interest since the early 1970s with the advent of ARPANET [3], the predecessor of the Internet. Yet today, we find that the different optimal routing algorithms developed over the last 40 years are seldom implemented. Instead, distributed link-state routing protocols like OSPF/IS-IS that support hop-by-hop packet forwarding are the dominant intradomain routing solutions on the Internet.

The driving force behind the widespread adoption of link-state, hop-by-hop algorithms has been their simplicity—the main idea is to centrally assign weights to links based on input traffic statistics, flood the link weights through the network, and then locally forward packets to destinations along shortest paths computed from the link weights. As our communication networks have grown rapidly in size and complexity, this simplicity has helped OSPF eclipse extant optimal routing techniques that are harder to implement.

However, the obvious tradeoff has been lost performance. For instance, due to the poor resource utilization resulting from OSPF, network administrators are forced to overprovision their networks to handle peak traffic. As a result, on average, most network links run at just 30%–40% utilization. To make matters worse, there seems to be no way around this tradeoff. In fact, given the offered traffic, finding the optimal link weights for OSPF, if they exist, has been shown to be NP-hard [4]. Furthermore, it is possible for even the best weight setting to lead to routing that deviates significantly from the optimal routing assignment [4].

Our goal in this paper is to eliminate this tradeoff between optimality and ease of implementation in routing. The result is Hop-by-hop Adaptive Link-state Optimal (HALO), a routing solution that retains the simplicity of link-state, hop-by-hop protocols while iteratively converging to the optimal routing assignment. To the best of our knowledge, this is the first optimal link-state hop-by-hop routing solution.

Not surprisingly, there are multiple challenges to overcome when designing such a solution. Before getting into them, we define the following important recurring terms for ease of exposition.

Hop-by-hop: Each router, based on the destination address, controls only the next hop that a packet takes.

Adaptive: The algorithm does not require the traffic demand matrix as an explicit input in order to compute link weights. Specifically, the algorithm seamlessly recognizes and adapts to changes in the network, both topology changes and traffic variations, as inferred from the network states like link flow rates.

Link-state: Each router receives the state of all the network's links through periodically flooded link-state updates and makes routing decisions based on the link states.

Optimal: The routing algorithm minimizes some cost function (e.g., minimize total delay)

B. First Test

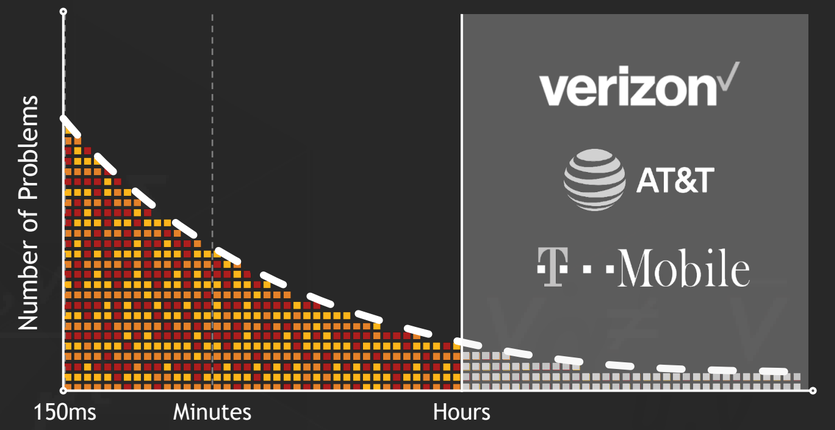
However, as a potential counterexample to this interpretation, it is possible to suggest some version of the scenario described in Fig. 1(b). Here, there is traffic demand of rate r from router A to router C. The initial splits at router A are α_m along an intermediate price link with price w_m and α_u along the more expensive route with price $w_B + w_t$, assuming $\alpha_t = 1$ initially. The relationship between the initial link prices are assumed to be $w_t > w_m > w_s + w_B$, i.e., link (A, B) is along the shortest path from A to C, but B also has the most expensive way to reach C. The concern is that router A shifting traffic from the intermediate price link to the link with price w_B might result in the cost increasing as router B initially routes traffic only through the most expensive link ($\alpha_t = 1$). However, because router B decreases α_t and increases α_s (in conjunction with the changes at router A), the total cost does in fact decrease. More precisely, the cost derivative can be calculated as follows:

$$\dot{\Phi} = -r \times \frac{\delta}{r} \times w_m + r \times \frac{\delta}{r} \times (w_B + w_t) - r_B \times \frac{\delta}{r_B} \times w_t + r_B \times \frac{\delta}{r_B} \times w_s = -\delta(w_m - w_B - w_s) \leq 0$$

where r_B is the incoming rate to C at B (superscript dropped for

Corrected in Hours

TELCO CONTROL

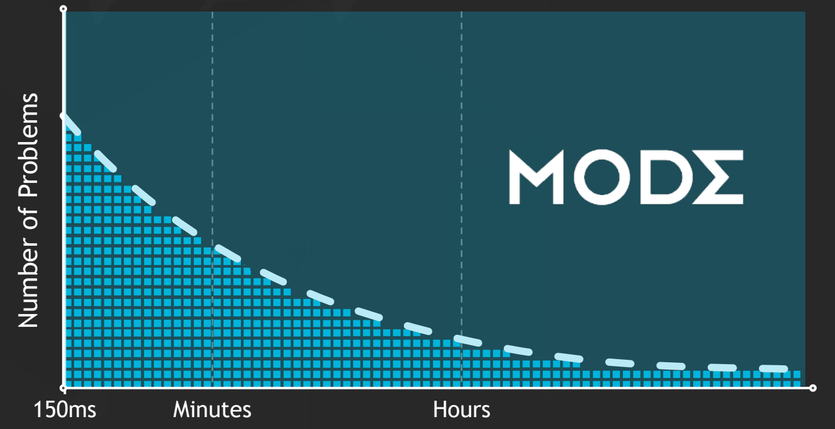


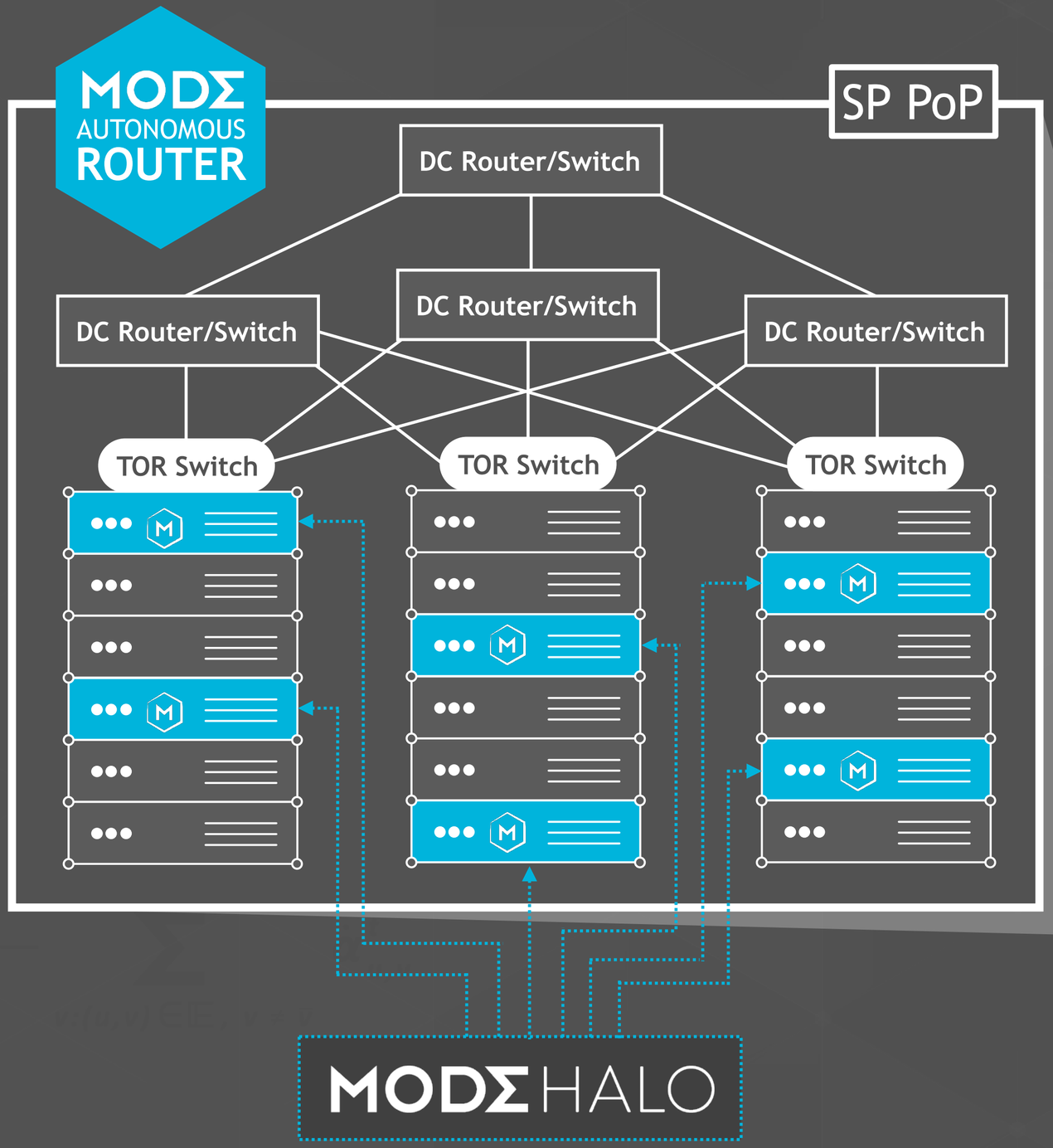
Corrected in Minutes

WEBSCALE CONTROL

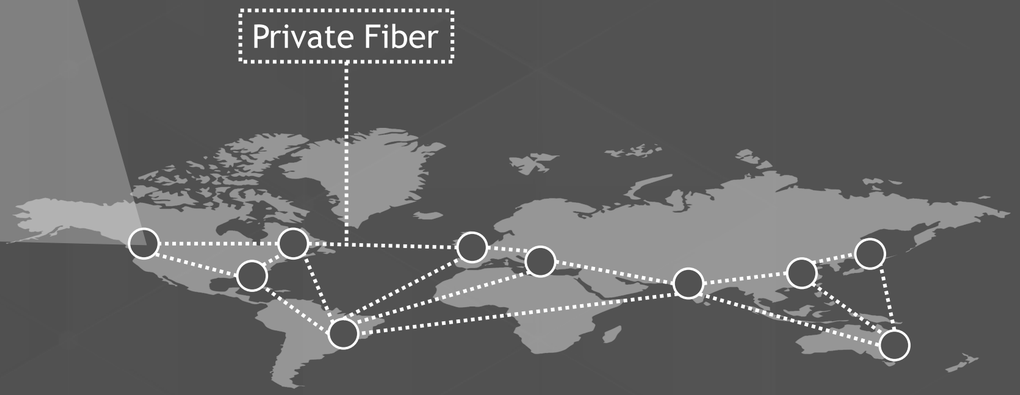
Ideal Every Moment

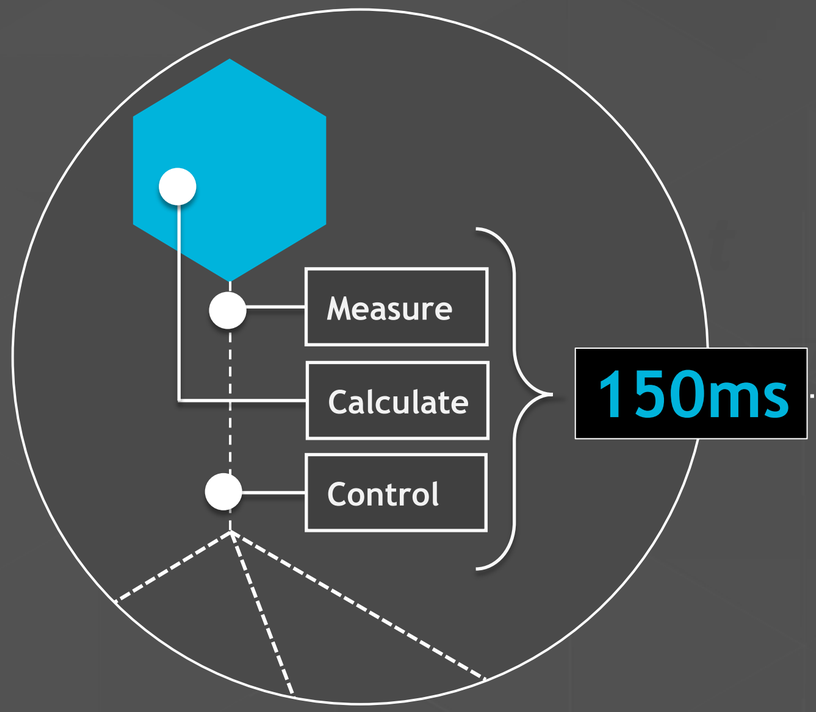
MILLISECOND CONTROL





The Mode software stack transforms COTS servers in network operator PoPs into optimally performing virtual routers, with comprehensive monitoring and management capabilities.

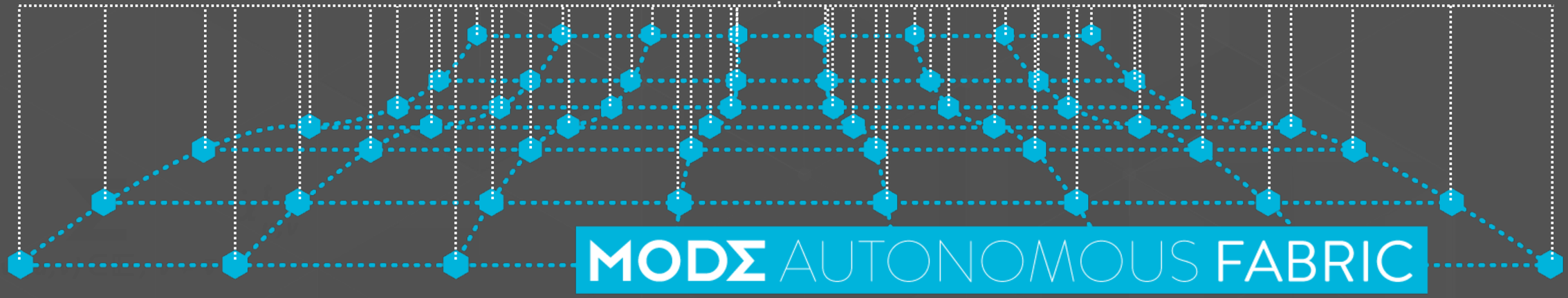




150ms

In Parallel

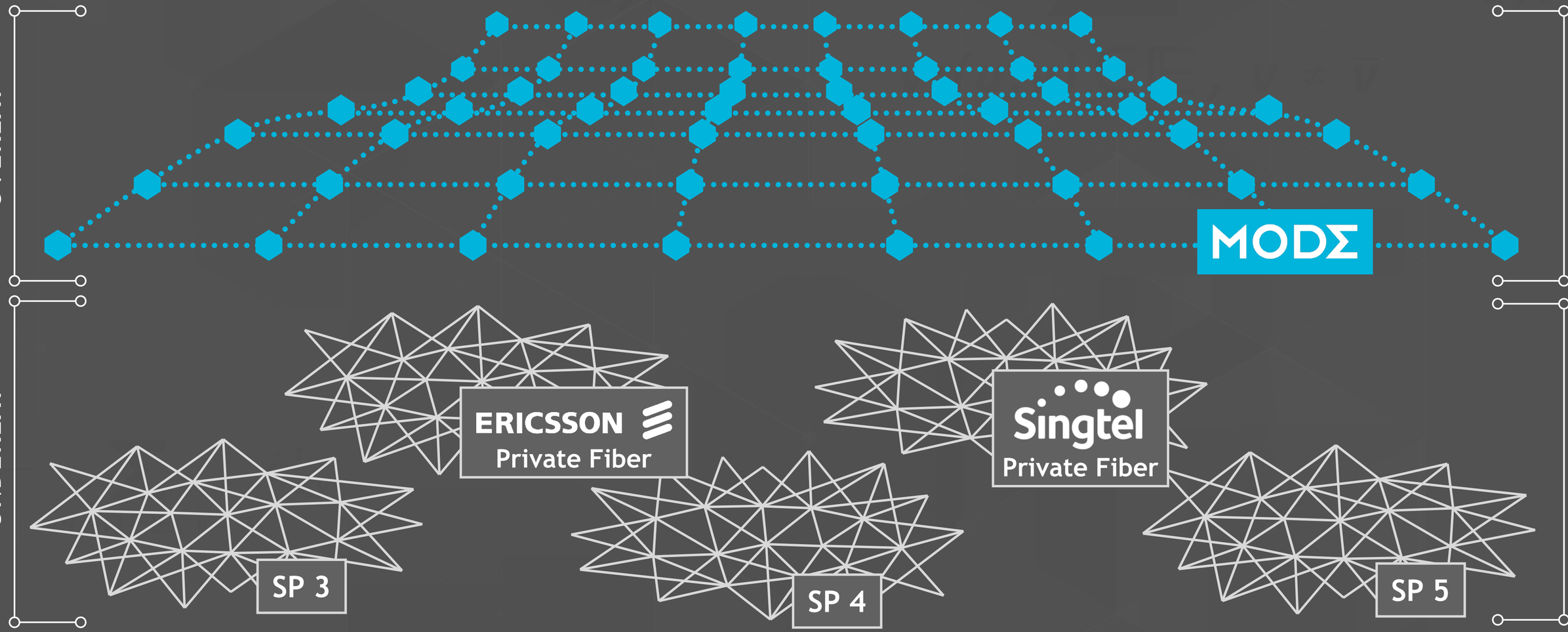
Autonomous, optimal routing with global reaction times under 150ms results in a game-changing shift in the efficiency and economics of private networking at global scale.



MODΣ SD-CORE

NaaS

We combine this overlay with private fiber from Ericsson, Singtel, and others, and offer it as a NaaS: **MODΣ** SD-CORE.



MODE SD-CORE

For SaaS Providers

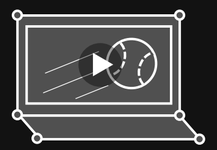
Remote work, distance learning,
interactive streaming, and government


UC/VoIP

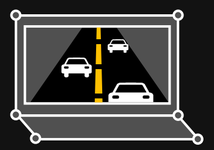

Cloud Collaboration


Distance Learning


Video Production


Live Event Streaming

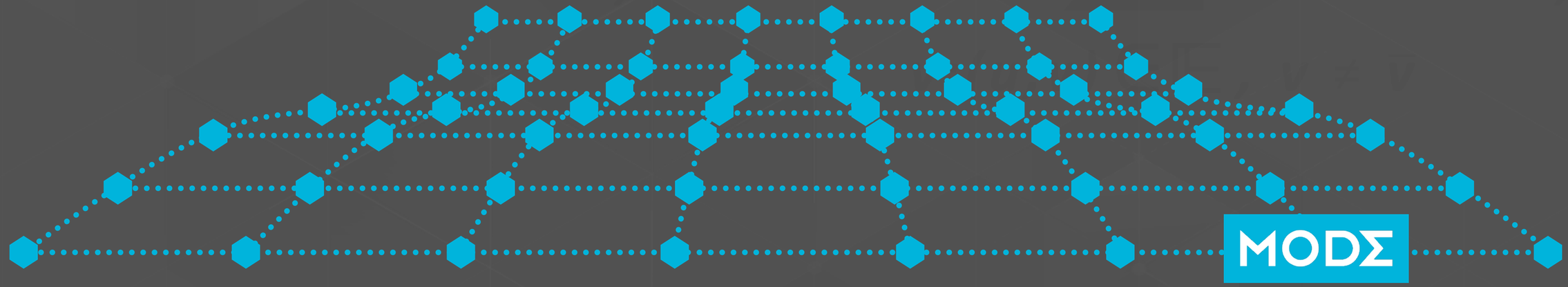

Government

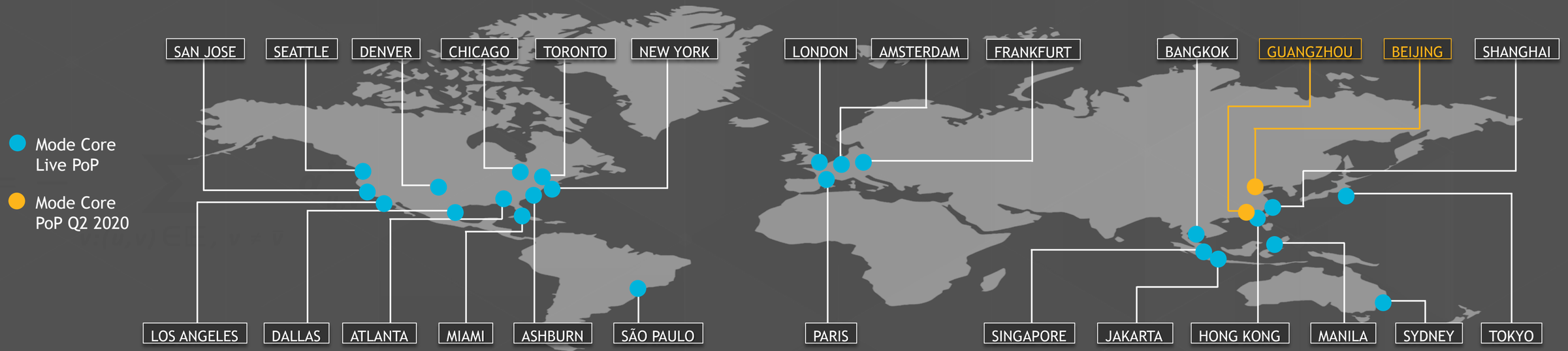

Game Streaming


IoT

MODΣ SD-CORE

MODΣ SD-CORE provides worldwide connectivity with 25 PoPs that span Shanghai, San Francisco, London, Sao Paulo, and beyond.



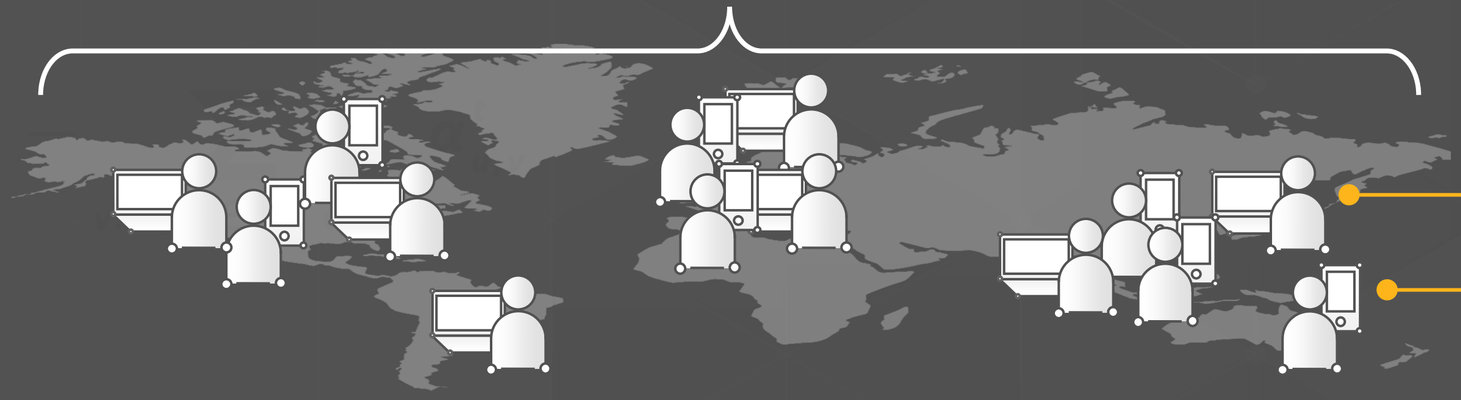


MODE SD-CORE for SaaS Providers

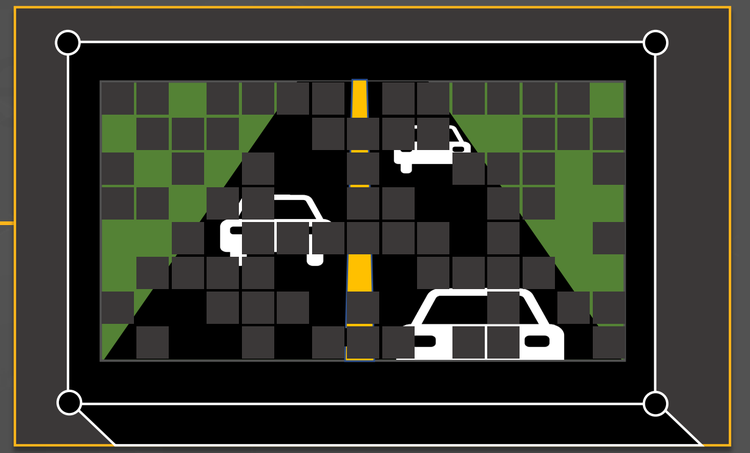
Internet Core Insufficient

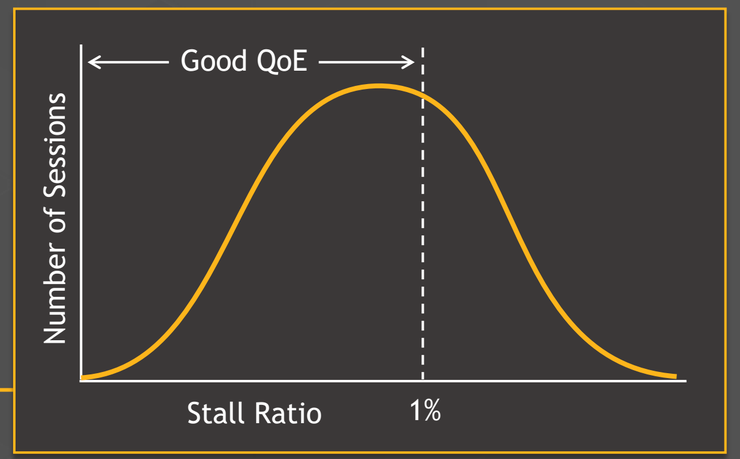
= SaaS Provider Server Locations

Internet Core

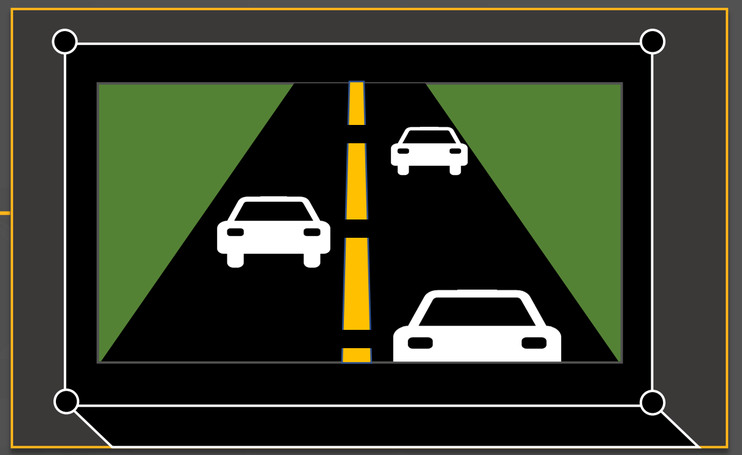


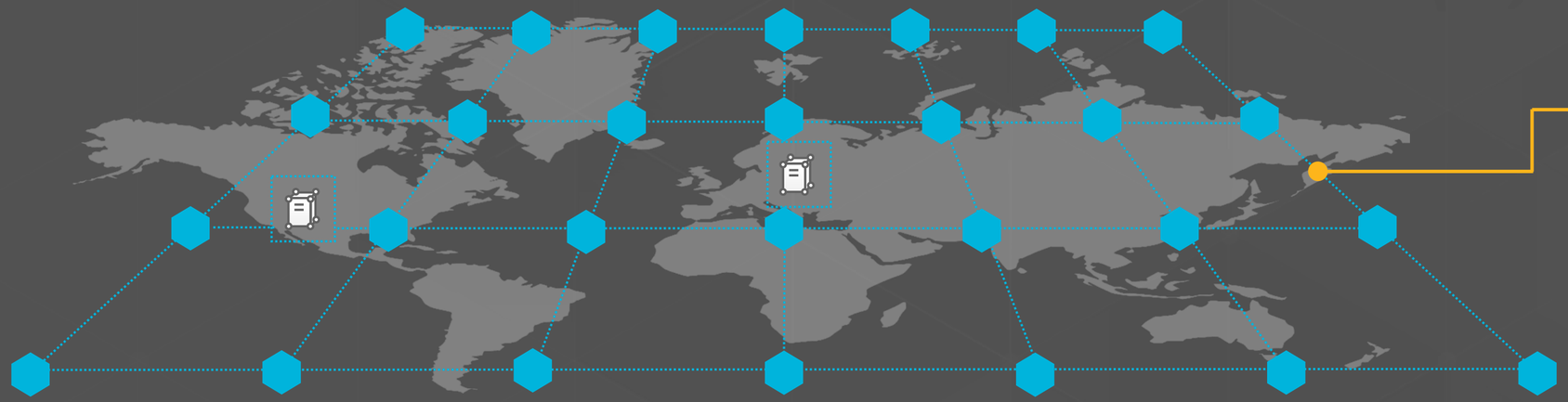
High OpEx



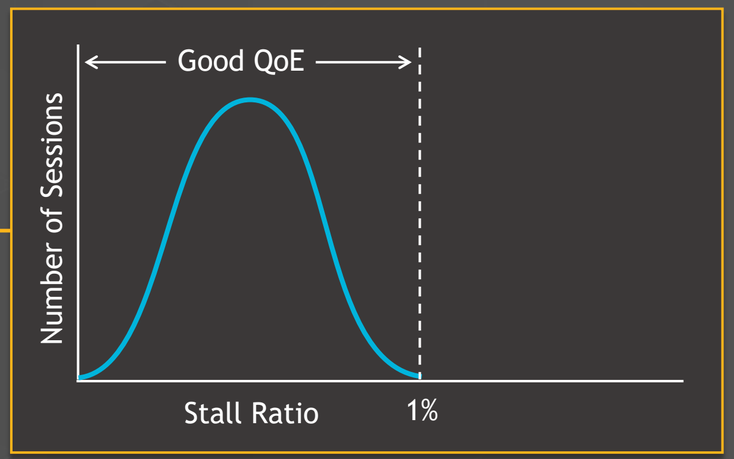


SaaS with MODE SD-CORE
Great QoE, Low OpEx





Dramatically Reduce OpEx

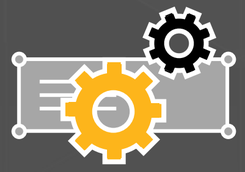


THE END OF NETWORRY™

THE IDEAL NETWORK AT EVERY MOMENT™

Performance & QoS

MultiCloud & SaaS QoE



Resource Optimization

Data Security & Compliance

Mobile QoS & Security

Scale & Cost

Edge-to-Edge Millisecond Control

MODE