DxOdyssey for IoT

Secure Software Defined Perimeter

Bi-Directional Connectivity for IoT

Introduction

The Internet of Things (IoT) is poised to irreversibly alter the contemporary data ecosystem with the promise to increase businesses' profits and improve customers' lives by collecting and extracting information generated by devices at the edge—the network boundary outside the datacenter and cloud.

IoT devices come in many forms: industrial robots on a factory floor, power sensors on streetlights, ovens in a restaurant, or a heart rate monitor embedded in a smartwatch. If accurately collected and analyzed, data generated on these devices can lead to important insights that bring health improvements to consumers, enhanced safety to workers, and cost savings to businesses and cities.

Let's consider the example of smart cities. Smart cities are on the rise—in a really big way. According to Microsoft, smart-city initiatives—which can be defined as cities that rely on Internet of Things (IoT) sensors to obtain data that's then mined to guide management of city services and resources—account for nearly a quarter (23 percent) of the world's IoT projects.

As the number of smart cities mushrooms, these hyperconnected urban areas are becoming increasingly critical to how seamlessly cities are able to operate. This is an important point to grasp, since cities serve as the linchpin for most of the world's data generation, as well as the majority of all energy consumption. What's more, most of us live in cities. The UN reports that just over half (55 percent) of the world's population makes a city their home—a figure that the UN predicts will rise significantly (close to 70 percent) in the next 30 years.

Here are some additional stats to impress upon you the importance of our urban areas in general, and smart city growth in particular:

- **Forbes reports that by 2025,** we'll be looking at approximately 80 billion devices that are smart devices.
- **By then,** our global cities may be cranking out up to 180 zettabytes of data.
- **In terms of energy production,** The World Bank reports that our urban meccas already gobble up to 80 percent of it worldwide.
- **Over the next decade,** cities will likely be responsible for close to three-quarters (74 percent) of global greenhouse gases, up from around two-thirds, or 67 percent, currently.

Smart cities are just one IoT application example. IoT applications are also being used in many different industries such as energy, agriculture, healthcare, and the future possibilities are endless.

Challenge with IoT Networking

To fuel these possibilities, new IoT edge compute platforms such as Microsoft's Azure SQL Edge are creating opportunities for innovation. These platforms are empowering developers to design solutions that ingest, store, score, and move IoT data at the edge, close to where data is being captured. But for all the benefits that these new IoT applications can deliver, IoT networking infrastructure is challenged. It's unwieldy and cumbersome in a way that often limits applications' potential for both the customer and developer alike.

The main problem with IoT networking is the security and configuration complexity of moving IoT data between edge devices, datacenter and cloud.

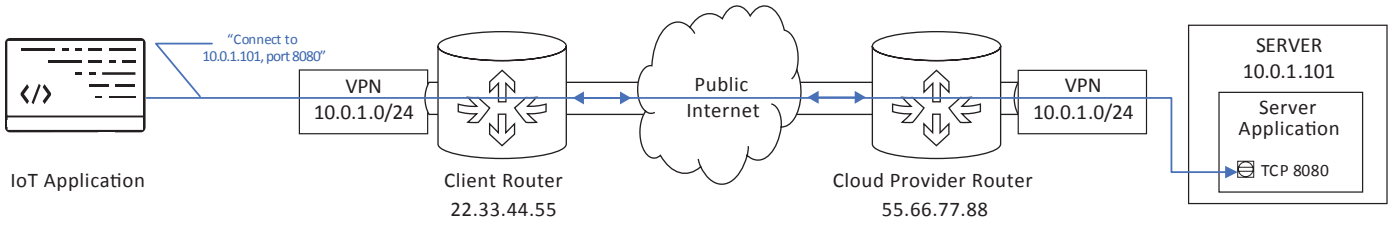
VPNs not designed for IoT

VPNs are about access to hosts and networks rather than specific application access. They extend an organization's network in ways that are open by default. These extensions can create vulnerability to lateral attacks, where a breach on one side of a VPN can more easily spread to the other side. Designed only for traditional perimeter enterprise security, they're entering obsolescence in today's IoT edge to datacenter and cloud environment. VPNs often require

» *Unlike VPNs, DxOdyssey works within existing on-premises, edge, and cloud network environments.*

dedicated hardware and are both complicated and costly to maintain for an option that merely widens the network surface area for attacks.

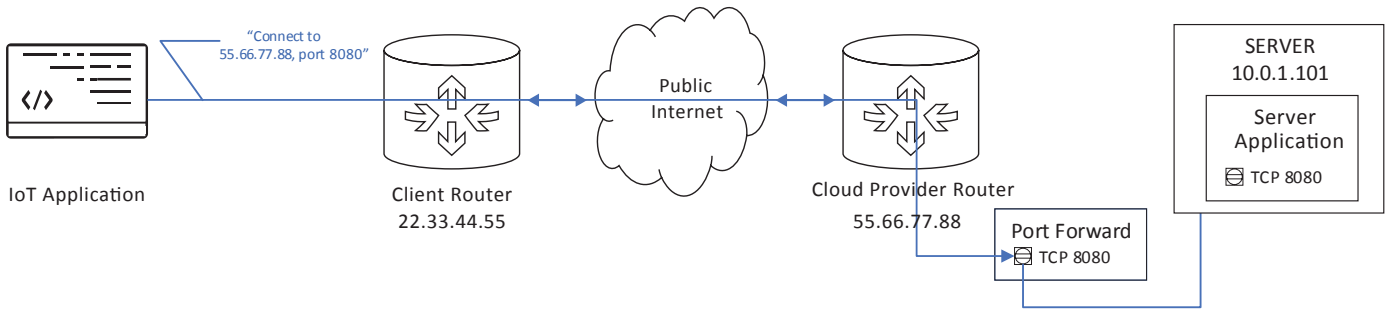
Figure 1: IoT edge device using a VPN to connect to a cloud server application



Open ports are an IoT security nightmare

The other common approach is to simply open up enterprise firewalls to enable direct access between IoT edge devices and datacenter and clouds. This method is decidedly worse than the former since it effectively opens networks to the entire Internet and every suspicious malefactor it contains.

Figure 2: IoT edge device opening ports to connect to a cloud server application



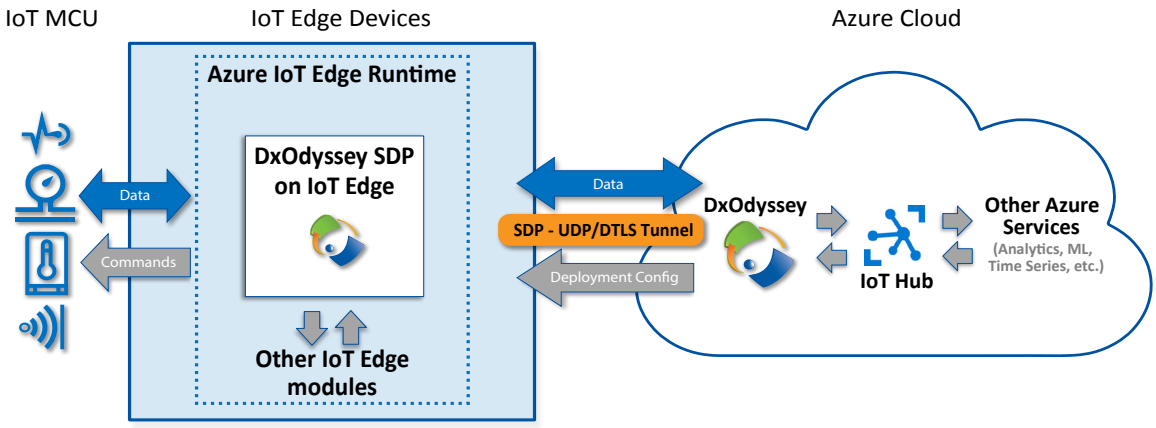
Safeguarding the Edge

A new network perimeter security model is needed for IoT edge devices. This new model would create a "zero trust" environment for IoT devices and applications. This means that edge devices would never use VPNs or expose public ports but instead use a Software Defined Perimeter (SDP) to manage application-level specific data transfers between IoT edge devices and various locations – whether datacenters, cloud, or other devices.

Introducing DH2i DxOdyssey for IoT

DH2i takes an innovative new approach to network connectivity by enabling organizations with its SDP Always-Secure and Always-On IT Infrastructure. DxOdyssey (DxO) for IoT extends this to edge devices, allowing seamless bi-directional access from edge devices to the datacenter and cloud.

Figure 3: DxO for IoT — Microsoft Azure SQL Edge use case



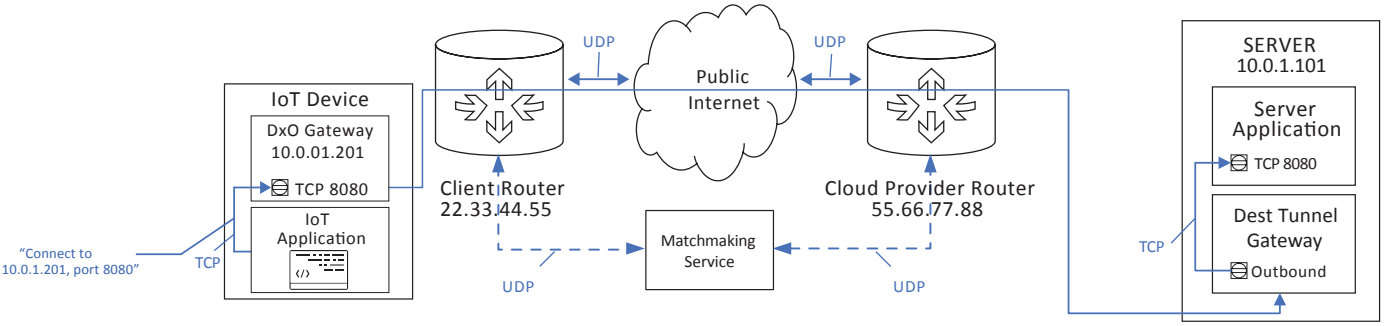
As shown in Figure 3, DxO for IoT gateway software is a container designed to run on x64 and arm64-based IoT devices located between microcontroller units (MCUs), such as sensors at the network edge. It's also designed to run on more powerful hosts in the datacenter or cloud. This SDP container is purpose-built for IoT use cases where bi-directional communication is desired between edge devices such as cameras and scanners, edge gateways that collect data from IoT sensors, and lightweight edge servers and datacenters and clouds. Once enabled, organizations can create secure, private application-level tunnels between devices and hubs without the requirement of a VPN or exposing public, open ports.

» *Minimum hardware requirements: Linux x64 and arm64 OS, 1 GB of RAM, 100 Mb of storage*

DxO for IoT is an emergent alternative that consistently proves cheaper, easier, and more effectual than both of those legacy approaches to IoT network security. DxO provides opt-in access to specific application micro-tunnels using enhanced user datagram protocol (UDP), which has randomly generated ports that render the tunnels basically invisible to cybercrooks, protected by Datagram Transport Layer Security and Public Key Authentication. This approach virtually eliminates lateral attacks and promotes a secure-by-default environment.

DxO for IoT further differentiates itself from VPNs by working within existing premises and cloud network environments, including NAT routers, usually without reconfiguration of the network components. DxO accomplishes this task using a highly available cloud-based matchmaking service to locate direct, discreet routes between IoT devices and sites through multiple layers of NAT routers.

Figure 4: IoT edge device using DxOdyssey to connect to a cloud server application



A deeper dive into the DxO for IoT solution elements illustrates how they combine to isolate IoT edge devices and applications between heterogeneous data settings for secure, discreet connectivity virtually no one, including the networks' participants, can detect.

DxO for IoT Gateway Software

The gateway software is the primary component of the DxO for IoT solution. It is installed on one or more IoT edge devices, datacenter hosts, and cloud hosts, and is formed into groups that span any set of environments that needs to be connected – IoT devices, on-premises, cloud servers, homes, and more. In figure 3, the DxO for IoT gateway software is running under the Azure IoT Edge Runtime on an IoT Edge device such as an office copier. DxOdyssey facilitates the ability for containers running on edge devices to securely connect with resources outside the local container network. Another DxO gateway is also installed on an Azure host.

The DxO gateways work together as a group to provide TCP micro-tunnel connectivity. An individual gateway can function as a tunnel origin, which accepts connections from IoT applications, or function as a tunnel destination that completes the connection to the server application. This way, the IoT application need not have access to the network where the server application is running, as long as the DxO gateway software is available on both.

When an IoT application connects to the DxO origin gateway on the TCP listening port specified by the tunnel definition, the connection request is forwarded to the appropriate destination gateway server over UDP which completes the connection using TCP.

Micro-tunneling enhances security by taking advantage of existing network isolation and providing access only to specific applications from specific points of origin.

Gateway Connectivity

DxO gateways communicate with one another using the User Datagram Protocol (UDP). Over UDP, DxO uses a condensed, all-inclusive communication protocol that provides security, authentication, liveness detection, configuration, and bearing of micro-tunnel traffic. Unlike most VPNs, DxO requires only a single UDP message channel between gateways. DxO is also able to take advantage of redundant network paths to provide seamless failover in the event of a partial outage.

» *DxO gateways communicate over UDP for security performance*

Communication between gateways is secured using the Datagram Transport Layer Security (DTLS) encapsulation. Similar to the more commonly used Transport Layer Security (TLS), DTLS provides a secure, authenticated, confidential message encapsulation between two parties. Unlike TLS, DTLS can tolerate operation over UDP which can re-order, drop, or duplicate messages en route. For DTLS, DxO uses security libraries bundled with the host operating system, which includes Microsoft's Schannel Security Service Provider on Windows and OpenSSL on Linux.

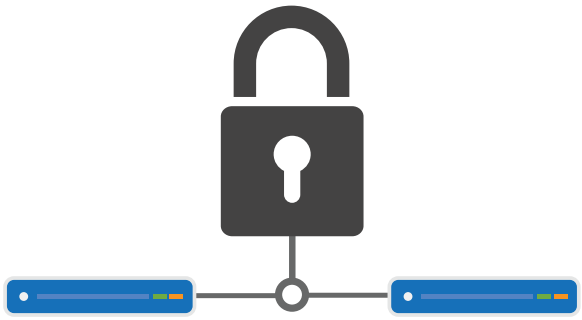
Cloud Matchmaking Service

DxO gateways are able to find others of their group and communicate directly over open, unencumbered IPv4 and IPv6 networks. However, when placed inside private networks or clouds behind NAT routers, gateway groups are unable to find other gateways directly. The matchmaking service remedies this by allowing gateways to identify whether they are behind NAT routers, and if so, to discover their externally-mapped UDP ports. The matchmaking service also allows gateways to communicate this information to other gateways of the same group to support direct communication thereafter.

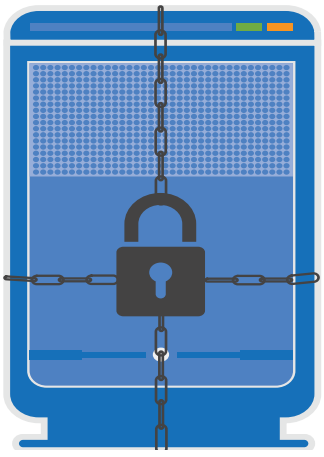
The use of the cloud matchmaking service allows DxO to be deployed across multiple isolated internal networks without having to add special configuration to the NAT routers.

Communication with the cloud matchmaking service is minimal, and is only used for finding other gateways of the gateway group. The matchmaking service is designed as a helper service, is not used to relay traffic between gateways, and is not permitted to affect group configuration.

» *Communication with the cloud matchmaking service is minimal, and is only used for finding other gateways of the gateway group.*



The DxO cloud-based matchmaking service delivers many key architectural advantages. Because it's simply a service in the cloud, scaling up or out to refresh gateways and/or to increase gateways' performance is as easy as hot-adding CPU, RAM, disks, and/or more gateways to the gateway group. The gateways periodically contact the matchmaking service and identify themselves to be matched.



» *A focus on data privacy means no user data is ever monitored or known to the matchmaking service*

Another key feature of DxO is its focus on data privacy. Once the matchmaking service identifies the gateways and introduces them to each other, it removes itself and is never in the data path of the gateways. Users' data are never monitored and known to the matchmaking service. In this regard, it's akin to an online dating service. It simply connects gateways (like a dating service connects daters); the gateways then create tunnels for application data exchange—much like daters using this web service for matchmaking and introductions—then talk to each other without involving any cloud-based service. This is a significant difference when compared with other software defined perimeter options that sit squarely in the path of the data they're transmitting, which can exacerbate regulatory compliance and security issues.

Micro-Tunnel Stream Transport

On each side of a micro-tunnel session, the client and server applications are connected via TCP to the DxO gateways. The gateways repacketize and forward the TCP traffic through their UDP message channels using an intelligent, lightweight, stream transport protocol. The stream transport protocol encompasses the critical functions of TCP including frame retransmission, selective acknowledgments, and congestion control.

» *DxO has no dependencies on the OS networking stack and does not require any special OS privileges*

The DxO micro-tunnel stream transport protocol is lightweight, highly performant, and implemented entirely inside the gateway software (i.e. layer-4). It has no dependencies on the operating system networking stack and does not require any special operating system privileges.

To configure a new tunnel between two gateways, the system sends a request to the destination gateway to create an internal micro-tunnel listener block. The listener block will be configured to listen on a randomly generated internal port and connect the destination target address – the target application for the tunnel. The system then instructs the origin gateway to configure a local TCP listener block, which allows IoT device to connect to and execute application requests. All IoT devices' requests received by the origin gateway on the local TCP listener block are packed and forwarded to the destination gateway over the UDP channel. When the requests arrive at the destination gateway, they are unpacked and sent to the target application over TCP. If a response or acknowledgement is needed by the target application, the data are sent to the destination gateway over TCP, which are then packed and forwarded to the origin gateway over UDP. When the response data arrives at the origin gateway, they are unpacked and delivered to the IoT device over TCP. For other IoT use cases, the tunnels can be configured so that the data flows to the edge device, or even both directions. Data transmission *from* premises or cloud to IoT edge works the same as described here once the tunnels have been defined.

From the remote endpoint's perspective, the only point of application contact is the local origin gateway. From the target application's perspective, its point of contact is the destination gateway. With such topology, the IoT device and the target application are never exposed to lateral access risks from external sources, even though the IoT device and the premises location or cloud are geographically apart.

Through micro-tunnel stream transport, all source data remain local. Those data resources never get on the public Internet; from the endpoint perspective, they simply access the origin gateway of DxO. The data exchange transfers through the public Internet and reaches the target application through DxO's technology. From the source application's perspective, the data never leave the local network – which is essential for mitigating any potential security hazards. This fact is one of the principal points of differentiation between DxO and most contemporary hybrid or multi-cloud security methods, including software defined perimeters. These tunnels enable organizations to effectively *access remote data as though they are local,* without any standard, open ports for scanners to detect.

In Summary

Overall, DxO for IoT SDP gateway software has a host of benefits when compared to VPNs or exposing open ports, foremost of which is its intrinsic security. IoT devices and premises/cloud applications are never exposed to the public Internet; they never even leave their local network when accessing their local tunnel gateway. With ease, the DxO gateway software and matchmaking service find paths through NAT routers, using dynamic, randomized ports, providing invisible connectivity buttressed by UDP, DTLS encryption and Public Key Authentication. This DxO for IoT module is a container purpose-built for IoT use cases. It also uses much less hardware than VPNs do, translating into lower total cost of ownership and greater adaptability. With this approach, any x64 and arm64-based IoT device or commodity x86 server can function as a gateway. In comparison, VPNs need dedicated appliances and routers with unavoidable RAM and storage limitations. The networking ease of DxO culminates in vastly superior performance boons for IoT networks. This lightweight software operates atop a user's existing network to create micro-tunnels and secure pipes while bypassing the plethora of networking components – and bottlenecks – that VPN data encounter.

When one considers these factors, it's much more than just a more cost effective, better, easier and faster methodology. It's the ultimate enabler of IoT zero trust security and that means making good on the promise of today's IoT world.

Try DxOdyssey for free:
dh2i.com/trial