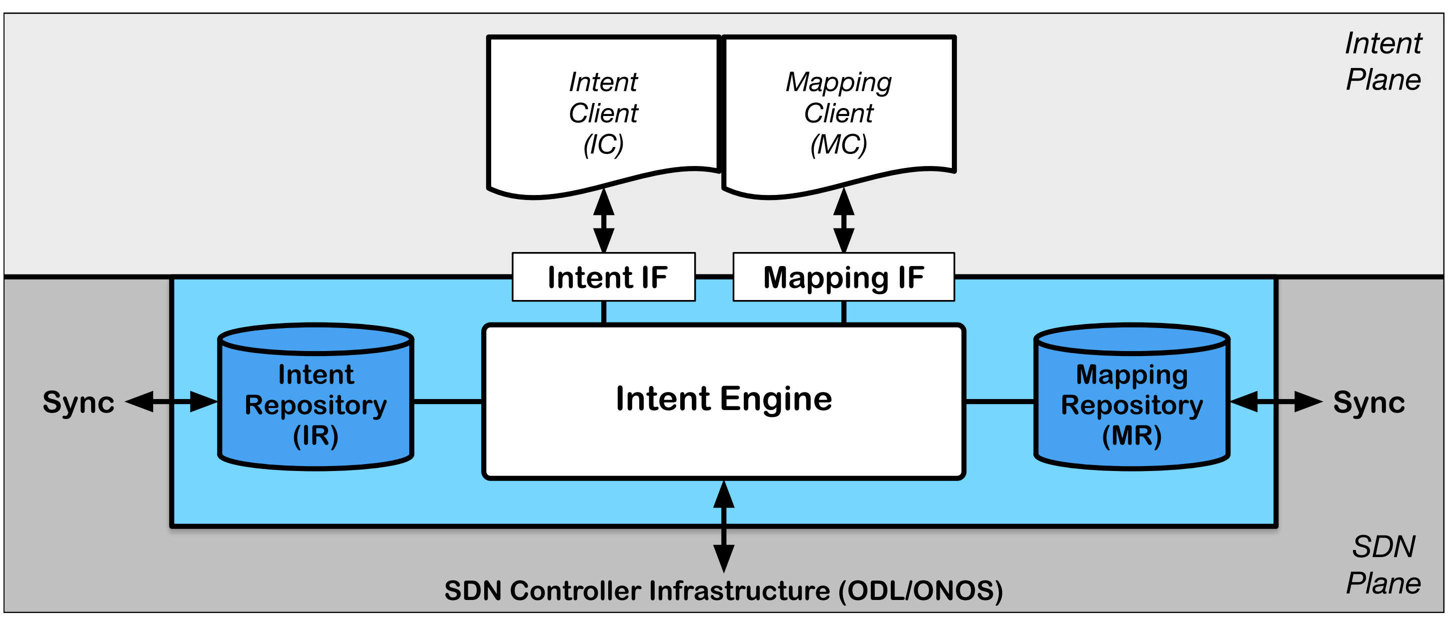
INTENT BASED NETWORKING

Last revised on Oct 26, 2016

The goal of the ONF/Atrium Enterprise project is to provide a working demonstration of intent-based networking following the vision described in the ONF document "Intent NBI – Definition and Principles" and continuing work initiated by the ONF/Boulder project. Our initial use case is intent-based control over voice and video calls across Microsoft's multi-site Skype-For-Business (SFB) infrastructure by delivering the associated media flows over the corporate WAN instead of the Internet under the expectation that the former offers a better quality of experience. We introduce the following intent-based networking concepts:

1. The Intent Repository (IR): A database that contains intent, provisioned by an Intent Client (IC) through a platform-independent Intent Interface (Intent IF) and distributed across the SDN infrastructure through some platform-dependent Sync interface. This database contains intent elements that are considered invariant and unconditional (focused on "what", not "how" nor "if").
2. The Mapping Repository (MR): A database that contains mappings, provisioned by a Mapping Client (MC) through a platform-independent Mapping Interface (Mapping IF) and distributed across the server infrastructure through a similar Sync interface. This database contains mappings between intent elements and the actual underlying resources. They are considered dynamic and could be conditional ("how" and even "if" to allow compromise).
3. The Intent Engine (IE): A module embedded in the SDN controller that reconciles the expressed intent and current mappings to instrument the physical infrastructure in order to achieve the stated aims.

The simple diagram below depicts the logical concepts. Note that it does not represent the actual software architecture which will combine the Intent and Mapping Interfaces into a single server interface - named the Intent NBI (North Bound Interface) - and where the Intent and Mapping Repositories are captured in a single database. In practice the intent and mapping clients are likely to be part of one application controller (also referred to as the call controller in the Microsoft use case).



The SDN controller infrastructure is targeted towards ODL and ONOS but other controllers are possible.

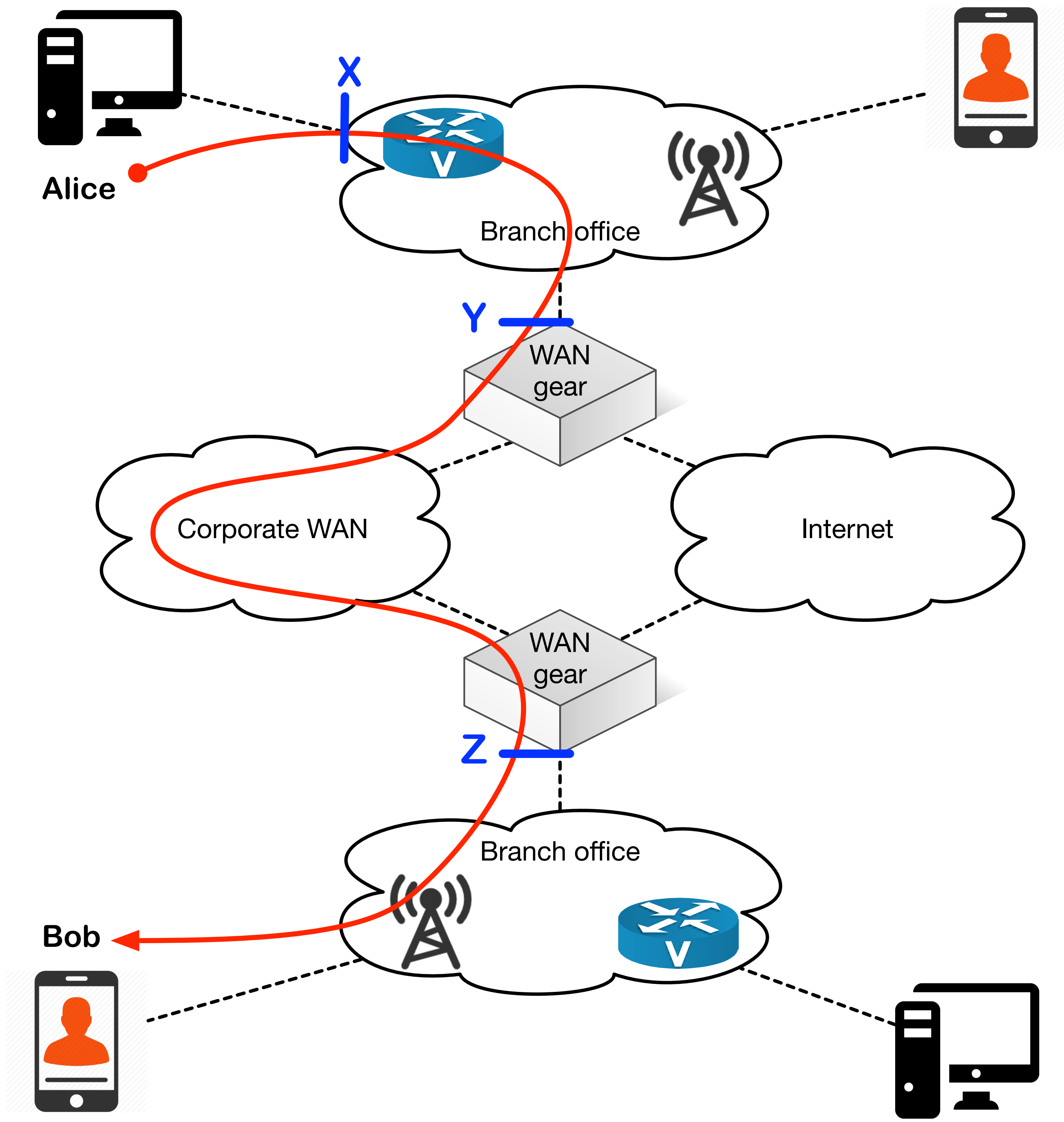
USE CASE OVERVIEW

The Microsoft use case involves two different network environments: The *application* network in the intent plane allows the initiation of voice and/or video calls using names and labels that the *physical* network in the SDN plane knows nothing about. A presence server resolves which of several devices a user is sitting in front of and, upon demand, the call controller sets up media flows passing specific transport, network and datalink address information to the SDN controller infrastructure.

By design, the call controller does not share information about user ID's, authentication, presence or policies with the SDN controller, which has no knowledge of the state of the application network. Similarly, the SDN controller shares nothing about its physical topology, attachment points, path selections, etc. with the call controller. It is the intent engine that determines where to enforce the intent-based behaviors, specifically by DSCP marking, leveraging a station registry in the SDN infrastructure that tracks where specific endpoints are attached to the SDN fabric.

The intent and mapping repositories will be synchronized across the customer's multi-site infrastructure so that the intent engine embedded in the SDN controller at each branch can independently decide how to impose these intent based behaviors on the local infrastructure. We explicitly strive to scale intent by avoiding the wholesome replication of SDN controller state across sites.

The illustration below shows a sample media flow in red from a caller (Alice) reaching its destination (Bob). For simplicity we ignore the largely symmetric media flow in the opposite direction. DSCP traffic marking will initially be applied at the WAN core boundary (Y) and could even be re-applied at boundary (Z) should the WAN transport modify the marking. In a later phase we could also mark the traffic at ingress into the branch network edge at boundary (X). Note that the pictured "WAN gear" could in fact reside inside the branch office as CPE gear or could be located inside the WAN cloud with only a transmission line terminating at the branch office. For our purposes, it does not yet matter as long as we are able to apply DSCP marking. We aim to do this by leveraging some OVS-based appliance - whether implemented in hardware or software – that is controlled by the local SDN infrastructure.



PROOF OF CONCEPT LIMITATIONS

Unfortunately, changing the routing of the media flows *after* they have been negotiated can lead to a disruption of communication because of differences in the actual IP addresses when traversing the WAN versus Internet: The latter is likely to include NAT devices and firewalls which are notorious for causing traversal problems (Refer to IETF RFCs on ICE, STUN and TURN for the complexities involved). For this reason, our initial use case will be limited to providing DSCP marking to allow the forwarding equipment to provide better quality-of-service treatment for the media flows but the ability to change the actual routing of the media flows will be explored once we have a better understanding of Microsoft's deployment plans for Internet connectivity at the various branch offices.

INTENT ELEMENTS

The Intent North-Bound Interface (Intent NBI) centers on three elements: Objects, object groups and modifiers.

Objects are named entities to abstractly represent users or equipment. Objects must be present in the mapping repository (as a key with zero or more values) before they can be referenced from the intent repository or intent NBI. In the above example "Alice" is an object that represents a participant in a call and who may in fact use multiple devices or locations as maintained in the mapping repository.

Object groups are named entities that contain other objects and/or object groups. They are simply collections that are defined in the intent repository. An example is a "High Quality Conference" object group to which Alice and Bob may belong to grant them a better experience.

Modifiers are named entities that can be associated with one or two objects or object groups to assign a desired behavior: The modifier is either "single-pronged" when applied to a singleton object or object group, or "dual-pronged" when applied to a pair but a modifier never has a span greater than two. The modifier is considered directional and any number of modifiers can be applied to a given singleton or pair. An example is a "High Importance" modifier that could be associated with the Alice object or with the High Quality Conference object group.

We will use a YANG model to define the Intent NBI as described above. Our goal is to start as simple as possible and augment the implementation as use cases are added.

OPERATION SEQUENCE

For the Microsoft use case, the following conceptual operations are performed during setup by the call controller:

1. The intent client creates an object group named High Quality Video (HQV) in the intent repository if it does not already exist.
2. The intent client associates the High-Importance modifier with the HQV object group to indicate that this object group needs preferential forwarding treatment. We assume that the intent engine supports the High-Importance modifier.
3. Upon demand, the call controller sets up a video conference call under a given name - for instance "VC#1" – which requires multiple unidirectional media sessions with given names – for instance "S#1", "S#2", et cetera between the call participants.
4. The mapping client creates an object "S#N" in the mapping repository for each media session to correlate the session object to the corresponding 5-tuple of the actual media flow, specifying the IP address, port and protocol, and the desired DSCP marking.
5. The intent client creates an object group "VC#1" in the intent repository for the corresponding video conference and adds all related session objects "S#1", "S#2" et cetera" to this object-group.
6. The intent client adds the "VC#1" object group to the HQV object group to indicate that preferential forwarding treatment is requested.
7. The intent engine determines that the media flows for the video conference call "VC#1" should be stamped with the requested DSCP marking. Note that the intent engine in each branch office uses some station registry to resolve which media flows originate in and then leave the local branch office network and thus require DSCP marking. For this we aim to leverage the capabilities of the SDN infrastructure.

When a call or call leg needs to be removed, these instructions are executed in reverse order, taking care not to prematurely remove any object or object-group definitions and leave behind dangling references. For instance, the session "S#N" needs to be removed from the object group "VC#1" before the session object itself is removed.

DISTRIBUTED OPERATION

The advantage of using intent is that its rules are compact, independent and easily distributed. Each branch offices in the Microsoft use case can have its own SDN controller infrastructure - a "cluster" – and do not need to exchange detailed network state information. Only the intent rules are synchronized across these clusters in a "cluster of clusters" hierarchy.