# Topology Aware System

When you are using an Overlay Network and all of the components of your system are in One Region, you can be tempted to view the group of datacenters as one system. Very often, this will just work, until it doesn't. This is due to the fact that Peter Deutsch' Fallacies of Distributed Computing are more visible on WAN connections than inside local area networks. Ignoring the topology of the network simplifies systems, but causes needless traffic over WAN connections, risking running into said fallacies sooner.

Therefore,

**Make your systems topology aware, so that as much traffic as possible is kept within a datacenter. Having your Service Registry be topology aware so that local instances of services are returned first on a lookup is a good start.**

One of Deutsch' Fallacies is "latency is zero". When these were written, the most prevalent network probably was Ethernet in either 10base2 or 10baseT physical configurations. Even though 10baseT looked like a star topology, the actual repeaters in use were hubs, basically replicating the coaxial bus of 10base2 inside them; CSMA/CD was used in both, meaning that especially on loaded networks, latencies were widely variable. As latency drives a large number of performance metrics, from partition sensitivity to the number of requests per second that can be made over a logical connection, reducing latency as much as possible is of paramount importance.

"Latency is zero" has become less and less important over time, as traffic inside a datacenter is now switched, fiber connections between racks of machines essentially operate at light speed, and intermediate routers only add miniscule delays. Round-trip times are measured in tens to hundreds of micro-seconds and one can get very far by essentially ignoring the network.

However, when adding multiple datacenters, one is back in the '90s again. Latencies are widely variable, varying between the tens to hundreds of milli-seconds, and ignoring the network when crossing the WAN will quickly lead to problems. Often, testing is done during "nice Internet weather", were everything will look nice and latencies are basically 50% of lightspeed, but there is pretty much a guarantee that no matter how traffic is routed between datacenters, there will be bad days. Or weeks. Cable repairs in remote areas are hard, damaging them is not.

In order to be successful in a multi-datacenter setup, it's therefore very important to keep traffic flows inside the datacenter as much as possible; in fact, WAN traffic should be the exception and every time this happens, there should be a clear and good reason for it. Some examples of "good" WAN traffic are cluster state information and data replication; some examples of "bad" WAN traffic are webservers talking to remote databases or microservices talking to other microservice instances in remote datacenters.

Systems need to be made aware of where services reside, and need to have a strong preference for local instances; in fact, when critical systems (like databases) are not available in a datacenter, it is probably preferable to have client systems fail in that datacenter as well rather than going over the WAN, stressing remote systems needlessly.