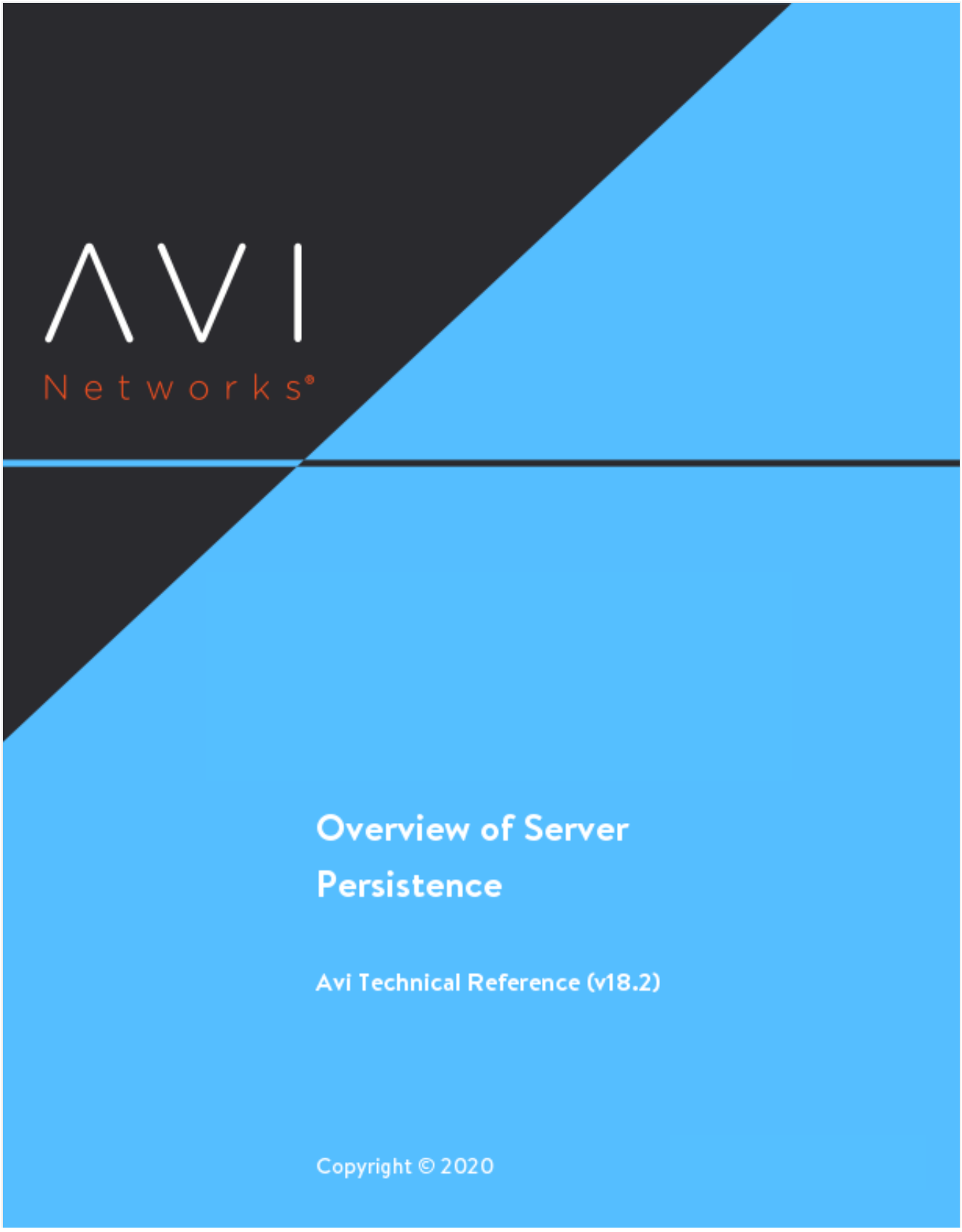
AVI
Networks®
Overview of Server Persistence
Avi Technical Reference (v18.2)
Copyright © 2020

Overview of Server Persistence

view online

A persistence profile governs the settings that will force a client to stay connected to the same server for a specified duration of time. This is sometimes referred to as 'sticky connections'. By default, load balancing may send a client to a different server every time the client connects with a virtual service or even distribute every HTTP request to a different server when connection multiplex is enabled. Server persistence guarantees the client will reconnect to the same server every time when they connect to a virtual service, so long as the persistence is still in effect. Enabling a persistence profile ensures the client will reconnect to the same server every time, or at least for a desired duration of time. Persistent connections are critical for most servers that maintain client session information locally.

All persistence methods are based on the same principle, which is to find a unique identifier of a client and remember it for the desired length of time. The persistence information may be stored locally on Vantage Service Engines, or may be sent to a client, such as through a cookie or TLS ticket. The client will then present that identifier to the SE, which directs the SE to send the client to the correct server.

Persistence is an optional profile, configured within Templates > Profiles > Persistence Profile. Once the profile is created, it may be attached to one or more pools.

Types of Persistence

Vantage may be configured with a number of persistence templates:

- HTTP Cookie: Vantage inserts a cookie into HTTP responses
- App Cookie: Vantage reads existing server cookies or URI embedded data such as JSessionID
- HTTP Header: Administrators may create custom, static mappings of header values to specific servers
- Client IP Address: The client's IP is used as the identifier and mapped to the server
- TLS: Persist information is embedded in the client's SSL/TLS ticket ID

Outside of the persistence profiles, two other types of persistence are available:

- DataScript: Custom persistence may be built using DataScripts for unique persistence identifiers
- Consistent Hash: This is a combined load balancing algorithm and persistence method, which can be based on a number of different identifiers as the key

Persistence Mirroring

Persistence data is either stored locally on Vantage Service Engines, or is sent to and stored by clients.

Client stored persistence, which includes HTTP cookie, HTTP header mapping, and *consistent hash*, are not kept locally on Service Engines. When the data, such as a cookie presented by the client, is received, it contains the IP address and port of the persisted server for the client. No local storage or memory is consumed to mirror the persistence. Persist tables may be infinite in size, as no table is locally maintained.

Locally stored persistence methods, which includes *HTTP* app cookies, *TLS,* client IP addresses, and DataScripts, Vantage SEs maintain the persist mappings in a local table. This table is automatically mirrored to all other Service Engines supporting the virtual service as well as the Controllers. An SE failover will not result in a loss of persistence mappings. To support larger persistence tables, allocate more memory to Service Engines and the SE Group > Connection table setting.