

Destination Identifier for CONNECT operation

August 16, 1999
Akihiro Shimura
CANON INC.

The SERVICE_ID parameter

- **The SERVICE_ID parameter is used in the following two places;**
 - **Destination identifier for the CONNECT request operation, and**
 - **Service descriptor for the SERVICE_DIRECTORY operation response.**
- **By sharing a common name string in both places, the service descriptor describes destination identifier by itself.**

The SERVICE_ID and Destination Identifier

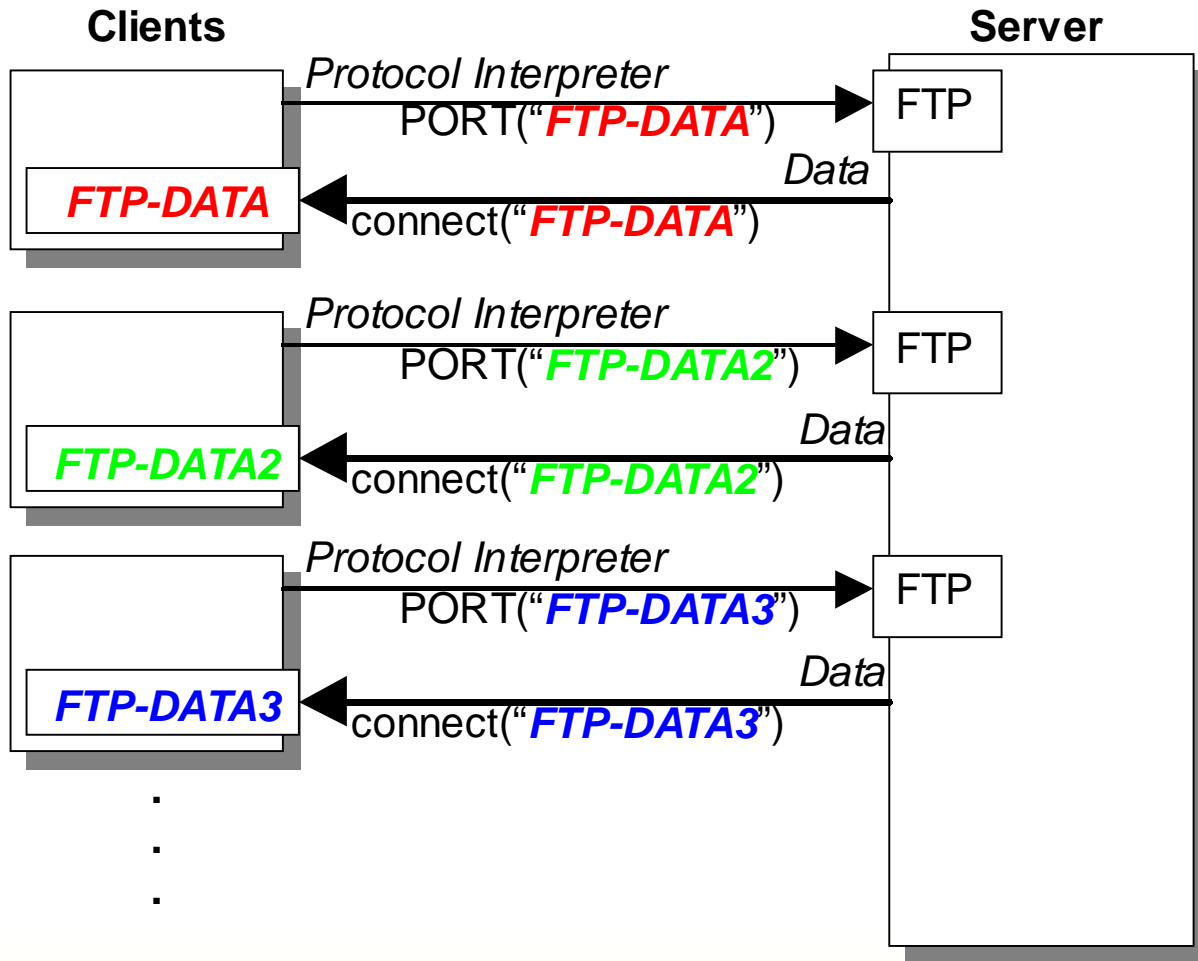
- **SERVICE_ID is**
 - a service name registered with a naming authority prior to use
 - *globally* unique within a name space of the naming authority, and *statically* valid
- **Destination Identifier**
 - uniquely identifies the listening service "entity" within a *local* login context at *connect time*.

Problems

- More than one service "entity" that has the same *global* registered name cannot co-exist within a *local* login context at the same time.
- The name needs to be *globally* registered prior to listen even if the name of the listening service "entity" only needs to be assigned uniquely in the *local* login context.
- *Global* name space will be exhausted if used for *local* identification.

Example (FTP)

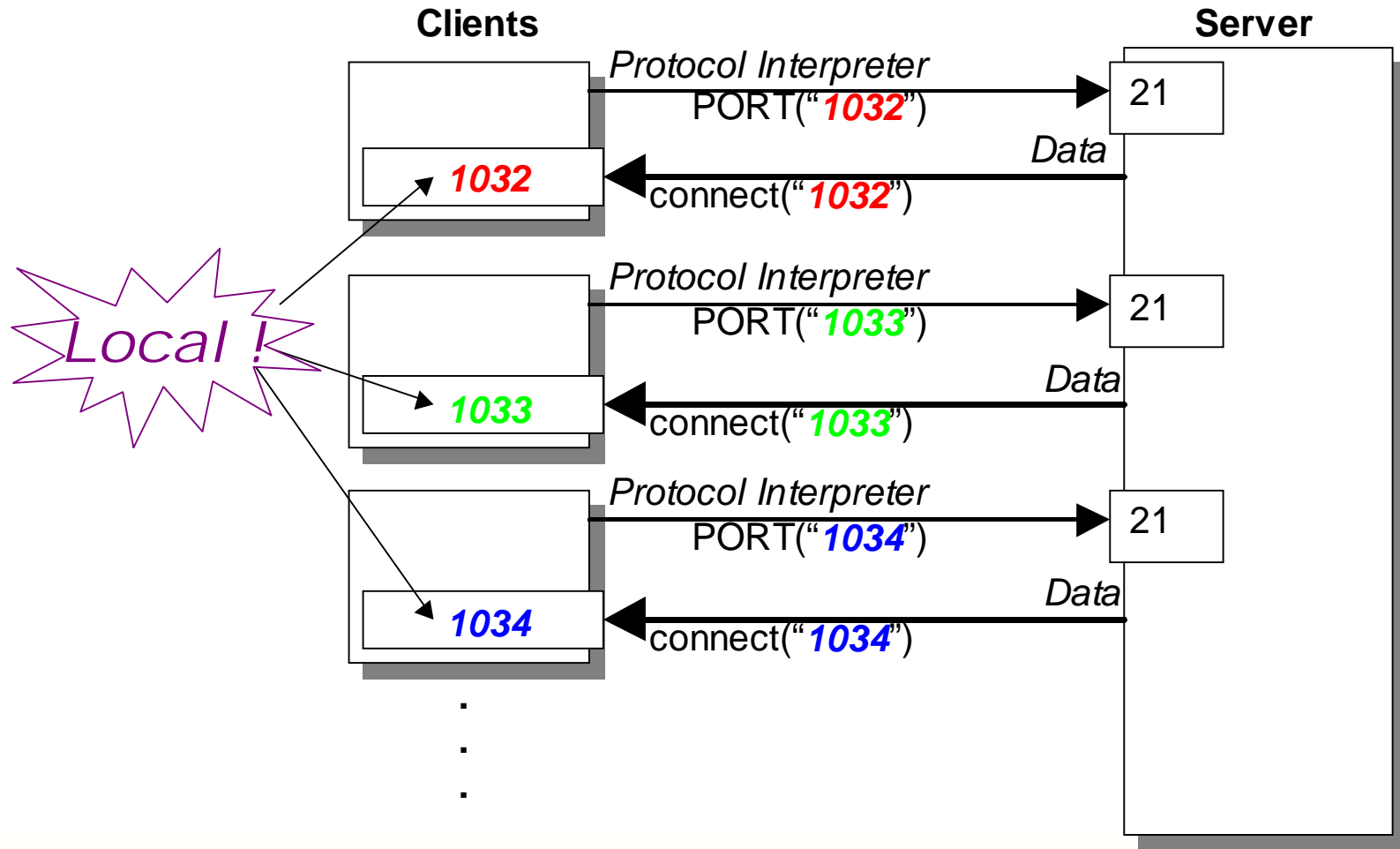
- Global Name Space*
- FTP
 - FTP-DATA
 - FTP-DATA2
 - FTP-DATA3
 - FTP-DATA4
 - FTP-DATA5
 - .
 - .
 - .



“DEST_ID” parameter

- **“DEST_ID” enables *local* assignment of destination identifier for CONNECT operation**
- **“DEST_ID” replaces “SERVICE_ID” parameter on the use of destination identifier for CONNECT operation**
- **“DEST_ID” is a number uniquely assigned to the service “entity” during its lifetime among all service entities within a fetch agent context (login) on a node.**

How the "DEST_ID" works



Conclusion

- Employing the *globally* registered service name as a destination identifier will make the application of the transport very restrictive, or otherwise, the global name space will be exhausted.
- By introducing "DEST_ID" that isolates destination identifier from *global* service name space, we will be able to eliminate unnecessary restriction to the application and/or impractical load to the naming authority.