

SPS

Snapshot Push Service

Technical White Paper

CS2 Network
Charlie

Date: 2018/01/05
Doc. Ver: 0.2

Agenda

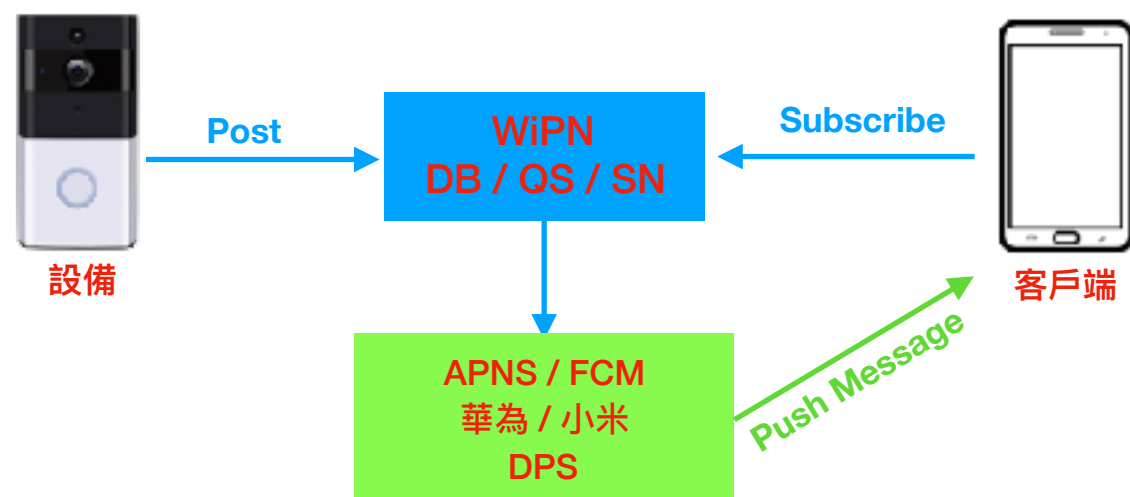
- Before Reading
- What is SPS
- Application Architecture
- About SPS Server
- SPS API

What is SPS

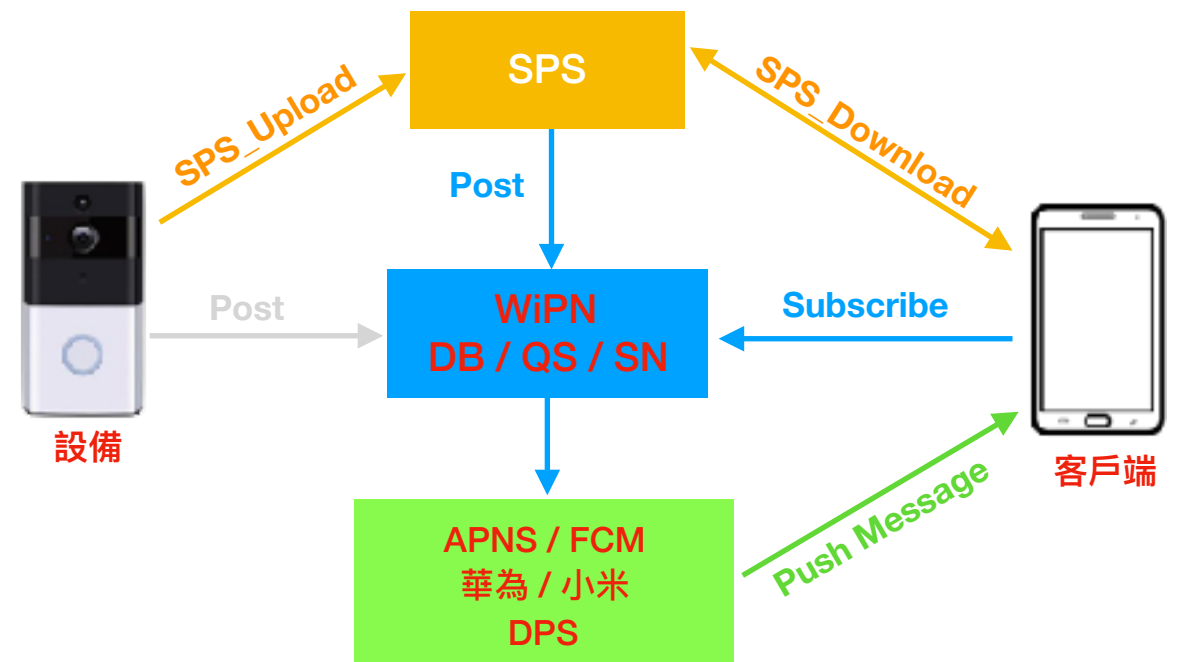
- SPS stands for Snapshot Push Service.
- SPS service includes: **SPS Server** (Linux x86,x64), **SPS API** (iOS / Android / Linux / Windows / osX / Embedded Linux / various RTOS)
- SPS support not only Snapshot, but also Video/Audio clip , normal file or any data, or even “Nothing”. By Nothing, it means SPS is just a agent of WiPN-Post.
- SPS is part of AiPN. SPS CAN NOT work alone without AiPN.

Application Architecture

Application Architecture of WiPN



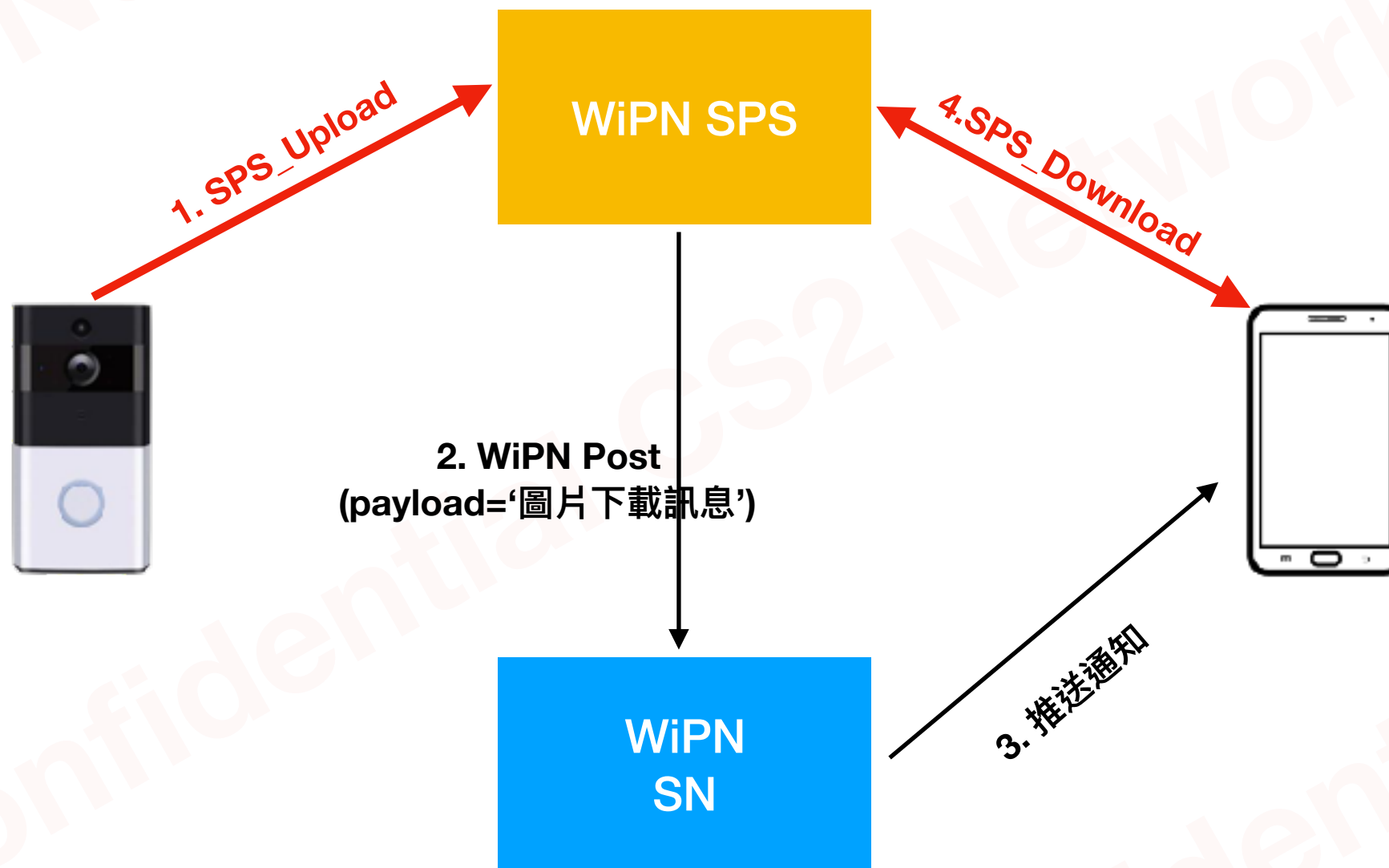
WiPN with SPS



About SPS Server

- SPS Server, a x86/x64 Linux program, host on a cloud server with fix Internet IP.
- SPS Server is a storage of Snapshot as well as a agent of WiPN-Post action.
- SPS will queue post action until it is successfully done or the Snapshot file is removed. (ps: Retry every second)
- SPS Server use TCP for snapshot upload and download.
- All data transmissions are proprietarily encrypted, and require a authorization (the **AuthWord**, defined in SPS Server's conf) verification procedure.
- Multipole SPS Server are preferred. So that the whole SPS service could be distributed and redundant.
- SPS Servers information, such as IP & Port, are provided by WiPN QS.
- SPS Server will not permanently save the Snapshot files. It's goal is to provide a temporary download buffer.

CS2 SPS API



- INT32 CS2_SPS_Upload(const CHAR *SPSInfo, const CHAR *AuthWord, const CHAR *DID, const CHAR *LNS, const UINT32 CH, const CHAR *PINFO, const CHAR *AINFO, const CHAR *SnapshotBuf, const UINT32 SSBufSize, CHAR *PostResultBuf, const UINT32 PRBufSize)
- INT32 CS2_SPS_Download(const CHAR *DLInfo, const CHAR *AuthWord, const CHAR *FileName, CHAR* SnapshotBuf, UINT32 *SBufSize, const CHAR FunCode)

SPS APIs

- **INT32 CS2_SPS_Upload(const CHAR *SPSInfo, const CHAR *AuthWord, const CHAR *DID, const CHAR *LNS, const UINT32 CH, const CHAR *PINFO, const CHAR *AINFO, const CHAR *SnapshotBuf, const UINT32 SSBufSize, CHAR *PostResultBuf, const UINT32 PRBufSize)**
 - **SPSInfo**: The Server information of SPS. This information is provided by QS of WiPN.
 - **AuthWord**: The authentication word. This must match SPS's setting otherwise SPS Server will not accept connection. AuthWord can be at most 16 characters. If exceeds 16 characters, only the first 16 characters are compared.
 - **DID, LNS, CH, PINFO, AINFO**: Information needed for Post action.
 - **SnapshotBuf**: The buffer of snapshot picture, Or the full file path name of snapshot picture. Or NULL.

If **SnapshotBuf** is NULL, you must use a thread to call SPS_FeedSnapshotData() to provide Snapshot data.
 - **SSBufSize**: The size of snapshot.

If the **SnapshotBuf** is a full file path name of snapshot (for example: /the/full/path/filename), **SSBufSize** must be 0xFFFFFFFF,.

If **SnapshotBuf** is NULL, **SSBufSize** tells the size of snapshot (if size is unknown, please enter 0xFFFFFFFF).

Otherwise, **SSBufSize** tell the size of **SnapshotBuf** in memory.
 - **PostResultBuf**: If PostResultBuf != NULL && PRBufSize != 0, CS2s3_Upload() will wait till post action finished and post result is stored in PostResultBuf. If PostResultBuf == NULL, CS2s3_Upload() will not waiting for post result.
 - **PRBufSize**: size of PostResultBuf. If PRBufSize is not big enough, the post result will be truncated.
- **INT32 CS2_SPS_FeedSnapshotData(const UCHAR *Buf, const UINT32 Size)**: To externally provide snapshot data for CS2_SPS_Upload() with parameter **SnapshotBuf** == NULL., if **Size** is 0, it means finish feeding data.

SPS APIs

- **INT32 CS2_SPS_Download(const CHAR *DLInfo, const CHAR *AuthWord, const CHAR *FileName, CHAR* SnapshotBuf, UINT32 *SBufSize, const CHAR FunCode)**
 - **DLInfo** The IP:Port information to download snapshot. DLInfo is inside push content.
 - **FileName**: the FileName of snapshot. SPS IP, Port and FileName is provided in payload of Push notify content.
 - **SnapshotBuf**: The memory buffer or the file path name for download snapshot from SPS Server.

If SnapshotBuf is memory buffer, you must declare SnapshotBuf with memory size > snapshot size, which is provided in payload of Push notify content.
 - **BufSize**: When calling, it specify the sizeof SnapshotBuf.

if *BufSize == 0xFFFFFFFF when calling, it indicated that SnapshotBuf is a file path name.

if *BufSize != 0xFFFFFFFF when calling, it indicated that SnapshotBuf is a memory buffer.

When returned, it is the actual size of SnapshotBuf downloaded.
 - **FunCode**: 0: Download only, 1: Download and Delete file, 2: Delete file only (don't download)
- **INT32 CS2_SPS_Break()**: to break executing of CS2_SPS_Upload and CS2_SPS_Download.
- **CHAR* CS2_SPS_GetAPIVersion(UINT32 *Version)**: To get information of CS2 SPS API.
- **void CS2_SPS_Initialize()**: to Initialize and start using of SPS_APIs.
- **void CS2_SPS_DeInitialize()**: to DeInitialize and stop using of SPS_APIs.

CS2 SPS API Return Code

- **Return value = 0: successfully.**
- **Return value > 0: network or socket error, please refer to definition of linux errno**
- **Return value < 0: API Error, as following**
 - -1: Invalid Parameter
 - -2: Invalid FunCode
 - -3: TimeOut (over 5 sec) waiting for Server's response
 - -4: AuthWord mismatch
 - -5: FileName doesn't exist
 - -6: Snapshot buffer size is too small
 - -7: User break, ie CS2_SPS_Break() is called
 - -8: The system memory is not enough for malloc
 - -9: Something wrong with 'DID'
 - -10: Server has closed tcp session, please look up Server Log for what happened.
 - -11: Local socket read timeout (5 sec)
 - -12: SPS_Upload is not calling
 - -13: The PINFO must be less than 4095 Bytes
 - -14: The AINFO must be less than 4095 Bytes
 - -15: The Snapshot must be less than 32 M Bytes
 - -16: The SPS_Upload function is temporarily disabled, due to too many Upload in last 10 minutes
 - -98: Server Error, Please check log of all CS2 SPS Servers, or contact CS2 FAE.
 - -99: Unknown Error, please contact CS2 FAE.

Snapshot Upload

- There are 3 ways to provide Snapshot for SPS_Upload().

1. Get Snapshot from a **file**, in this case:

SnapshotBuf is a file path & name string, for example: “/root/SPS/picture1”

SSBufSize is always 0xFFFFFFFF

2. Get Snapshot from **memory buffer**, in this case:

SnapshotBuf is a pointer to buffer where Snapshot is stored, for example: CHAR *pSnap;

SSBufSize is the size of **SnapshotBuf** in Byte

3. Get Snapshot by calling **SPS_FeedSnapshotData()**, in this case:

SnapshotBuf is NULL

SSBufSize is the size of **SnapshotBuf** in Byte or 0xFFFFFFFF

	SnapshotBuf	SSBufSize	Description
1	a file path & name string	0xFFFFFFFF	Get Snapshot from a file
2	a pointer to memory buffer	size of Snapshot	Get Snapshot from memory buffer
3	NULL	size of Snapshot or 0xFFFFFFFF	Get Snapshot by calling SPS_FeedSnapshotData()
4	NULL	0	Push Notification only, no Snapshot.

SPS Upload 的调用频率限制

- SPS_Upload() 的调用间隔会受限于过去10分钟内的SPS_Upload()调用的累计次数, 如下表所示:

10分鐘內的累計成功調用次數	調用間隔限制
0 ~ 5	0
6~10	3 秒
11~20	10 秒
21 次以上	30 秒

- 举例来说: 如果过去10分钟内 已经成功执行了 12次的SPS_Upload()了, 那第13次的调用时间至少必须与前一次间隔10秒, 如果在10内调用则会返回 -16 (ERRPR_SPS_UploadTemporaryDisabled)

Snapshot Download

- There are 2 ways to save Snapshot from SPS_Upload().

1. Save Snapshot into a **file**, in this case:

SnapshotBuf is a file path & name string, for example: “/root/SPS/picture1”

BufSize is always 0xFFFFFFFF

2. Get Snapshot from **memory buffer**, in this case:

SnapshotBuf is a pointer to buffer where Snapshot is stored, for example: CHAR *pSnap;

BufSize is the size of **SnapshotBuf** in Byte