INSUPC

(Intelligent Network Service User Profile Connectivity)

연동규격서

2019.03

요약

본 문서는 INSUPC 연동규격을 기술하였다.

\*) C/JAVA 서비스의 경우 별도 프로토콜 구현없이, 관리되는 cinsupc\_client, jinsupc\_client 라이브러리를 사용하여 SDK 형태로 사용할 수 있다.

변경이력

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 버전 | 날짜 | 변경내용 | 작성자 | 변경내역서 |
| 1.0.0 | 2011.03.10 | - AS간 연동 규격 최초작성 | 여경진 |  |
| 2.0.0 | 2019.03.31 | - BINARY 규격을 유지하면서 JSON 규격 추가 | 권용 | 업데이트 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1. 개요 4](#_Toc4974531)

[2. 시스템 구성도 5](#_Toc4974532)

[2.1. INSUPC와 AS 시스템 연동 구성도 5](#_Toc4974533)

[3. 인터페이스 구조 6](#_Toc4974534)

[3.1. 연결 방식 6](#_Toc4974535)

[3.2. 연결 구조 6](#_Toc4974536)

[4. 전문 구조 7](#_Toc4974537)

[4.1. 기본 구조 7](#_Toc4974538)

[4.2. 상세 전문 구조 및 기능 8](#_Toc4974539)

[4.3. 전문 구조 (버전 2.0) 14](#_Toc4974540)

[4.3.1. 기본 구조 14](#_Toc4974541)

[4.3.2. 시스템 인증 규격 14](#_Toc4974542)

[4.3.3. Heartbeat 규격 14](#_Toc4974543)

[4.3.4. Query 규격 14](#_Toc4974544)

[5. INSUPC와 AS 간 HEARTBEAT 구조 15](#_Toc4974545)

[5.1. 정상적인 HEARTBEAT 송수신 15](#_Toc4974546)

[5.2. 비정상적인 HEARTBEAT(No Response) 15](#_Toc4974547)

[5.3. 비정상적인 HEARTBEAT(No Request) 16](#_Toc4974548)

[6. INSUPC 이중화 정책 17](#_Toc4974549)

[6.1. 이중화 연결 구조 17](#_Toc4974550)

[6.2. INSUPC 이중화 정책 : Active-Active 구조 18](#_Toc4974551)

[6.2.1. AS 와 INSUPC 연결 전 19](#_Toc4974552)

[6.2.2. AS 와 INSUPC 연결시도 19](#_Toc4974553)

[6.2.3. AS 와 INSUPC 패킷 송수신 19](#_Toc4974554)

[6.2.4. AS 의 INSUPC 재연결 20](#_Toc4974555)

[6.2.5. AS 와 INSUPC 절체 20](#_Toc4974556)

[6.2.6. INSUPC 장애 발생 시 정책 20](#_Toc4974557)

1. 개요

본 규격서는 IMS기반 통화 응용/부가서비스를 제공하기 위해 응용서버(Application Server, 이하 AS)와 서비스를 위한 연동을 목적으로 인터페이스 규격을 정의한다. AS에서 호처리 진행중 필요한 서비스 정보를 AS🡪INSUPC, INSUPC🡪AS 의 흐름을 따라 인터페이스 한다. INSUPC와 AS간 연동 인터페이스는 TCP/IP를 기반으로 이루어진다.

본 규격서는 AS와 INSUPC간의 연동 인터페이스 규격을 명시한 것으로, 연동규격의 최소 요구사항이 기술되어 있고, 차후 서비스 확대시 추가 및 보완될 수 있다.

1. 시스템 구성도
   1. INSUPC와 AS 시스템 연동 구성도

AS와 INSUPC는 TCP/IP로 연결되고 AS는 Client기능을 INSUPC는 Server 기능을 한다. AS는 최초 연결 후정해진 규칙에 따라 INSUPC에 연결하고 호처리 시나리오 진행 중에 필요한 정보를 정해진 연동 규격에따라 INSUPC로 데이터를 요청하고 INSUPC는 AS의 요청에 따라 Database에서 데이터를 조회하여 AS에 전송해 주는 방식으로 동작한다.

ex) 통화서비스 Layer별 처리 시나리오

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

1. 인터페이스 구조
   1. 연결 방식

INSUPC-AS간 인터페이스는 TCP/IP 기반으로 연결하여 사용한다.

연결 구조는 최초 연결 후 인증 과정을 거치고 연결이 성립되면 연결세션을 유지하고 패킷을 송수신하는 방식을 갖는다.

AS는 TCP Client기능을 INSUPC는 TCP Server기능을 하며, 연결이 끊어졌을 경우 재연결을 시도한다.

AS는 연결된 세션을 통하여 연속적으로 패킷을 전송할 수 있고 INSUPC는 연속적으로 응답패킷을 전송할 수 있다.

하나의 AS는 INSUPC와 최대 n개까지 세션을 구성할 수 있다. 각 세션은 세션 아이디를 가지며 중복되는 세션이 발생할 경우 이전 세션은 Close 되고 새로운 세션으로 활동한다.

AS는 모든 INSUPC를 항상 연결된 상태를 유지한다.

* 1. 연결 구조



1. 전문 구조
   1. 기본 구조

Packet 구조는 그림1-1에서 보여지는 바와 같이 크게 헤더 부분(고정길이)과 실제 메시지 데이터의 내용을 담고 있는 Body부분(가변길이)으로 되어 있다

Body 부분에 포함되는 각종 파라미터에 대해서는 다음 섹션에서 설명한다

|  |  |  |  |
| --- | --- | --- | --- |
| **H**  **E**  **A**  **D**  **E**  **R** | MSG\_LEN (2byte) | 🡺 | Packet Body Size, 최대 4096byte |
| MSG\_CODE(1byte) | 🡺 | 메시지 코드 |
| SVCA(1byte)  Source Virtual Communication Address | 🡺 | 메시지 생성 주체 프로세스의 가상 주소값 |
| DVCA(1byte)  Destination Virtual Communication Address | 🡺 | 메시지를 전달받을 프로세스의 가상 주소값 |
| AS\_ID(1byte) | 🡺 | AS System ID |
| SESSION ID(30byte) |  | AS 에서 필요한 테이터 사용(보드,채널….) |
| SVC\_ID(4byte) | 🡺 | 서비스 타입 |
| RESULT(1byte) | 🡺 | 패킷 전달 결과 (APPENDIX A. 참조) |
| WTIME(17byte) |  | 패킷 전송 시간 |
| DUMMY(4byte) |  | 필요시 사용 |
| **B**  **O**  **D**  **Y** | Parameter count(1byte) |  |  |
| 1-Parameter type (1byte) |  |  |
| 1-Parameter length (2byte) |  |  |
| 1-Parameter data (m byte) |  |  |
| ... |  |  |
| n-Parameter type (1byte) |  |  |
| n-Parameter length (2byte) |  |  |
| n-Parameter data (m byte) |  |  |
|  |  |  |

[그림 1-1] Packet 구조

APPENDIX A.) Result Value

|  |  |
| --- | --- |
| 0x01 | 패킷전송 성공 |
| 0x10 | 패킷착신지 모듈을 찾지 못함 |
| 0x11 | 패킷 전송 실패 |
| 0x20 | 로그온 인증 거절(받은패킷그대로전송) |
| 0x21 | 과부하로 인한 폐기(받은패킷그대로전송) |

* 1. 상세 전문 구조 및 기능

이번 섹션에서는 database를 control하기 위해 공통 DB Server와 AS(Application Server) 간 주고 받는 Packet 기능 및 구조에 대해서 설명하며 구성은 아래와 같다

* Database Control Message
* Database Control Parameter
* Message와 Parameter의 대응관계
  + 1. Database Control Message

Database서버(DS)와 응용 서버(AS)간 데이터베이스 처리 수행을 위하여 전달될 수 있는 메시지에 대해서 [표 1]에서 정의하고 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| MSG CODE | MSG Name | symbol | 정 의 |
| 0x01 | DB\_QUERY\_REQUEST | dbq | DB Query 수행 요청 |
| 0x02 | DB\_QUERY\_RESPONSE | dbp | DB Query 수행 요청 결과 |
| 0x03 | DB\_ACCESS\_REQUEST | dbaq | DB Access 인증 요청 |
| 0x04 | DB\_ACCESS\_RESPONSE | dbap | DB Access 인증 요청 결과 |
| 0x05 | DB\_NETTEST\_REQUEST | dbnq | AS DB 간 HEARTBEAT Request |
| 0x06 | DB\_NETTEST\_RESPONSE | dbnp | AS DB 간 HEARTBEAT Response |
| 0x07 | DB\_STATUS\_REQUEST | dbsq | DB 상태정보를 AS에 전송 |
| 0x08 | DB\_STATUS\_RESPONSE | dbsp | DB 상태정보 전송에 대한 응답 |
| 0x09 | DB\_QUERY\_REQUEST\_ACK | dbqak | DB Query 수행 요청에 대한 수신 Ack  Header 만전송, SVCA, DVCA 만 바꾸어서 전송 |

<표1> Media Control Message Type

<표 1>에서 정의한 메시지들은 다음에서 정의하고 있는 database control parameter 들의 조합을 통해 database query 처리 수행 및 수행 결과를 알려준다.

* + 1. Media Control Parameter

🡪 아래 <표 2>는 database 를 control 하기 위해서 메시지 내에 포함되어서 사용되는 파라미터들에 대해서 설명하고 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter**  **Type (hex)** | **Parameter**  **Name** | **Parameter**  **symbol** | **내 용** |
| 0x01 | DB\_OPERATION\_ID | doi | Pre-develop된 db 수행 모듈 id |
| 0x02 | DB\_OPERATION\_NAME | don | Pre-develop된 db 수행 module name  DB Table 에서 Name + OP\_ID 값을 갖고 관리하고, DB내부에서 OP\_ID 기준으로 처리한다. |
| 0x03 | SQL\_INPUT | si | 위에서 언급한 db 수행 모듈이 수행되기 위해서 필요한 input data로 db 수행 모듈안에 구현되어 있는 sql query 문에서 사용하는 database field 의 변수값으로 사용된다. |
| 0x04 | SQL\_OUTPUT | so | DB 모듈 수행 후 얻어지는 output data로 시나리오에서 필요로 하는 DB Table/field에 저장된 값이다. |
| 0x05 | SQL\_RESULT | sr | DB 모듈 수행 처리 결과  (SUCCESS/FAIL 등) |
| 0x06 | DB\_STATUS | st | 공통 DB 상태  Active, Standby, Block, DR |
| 0x07 | DB\_LOGON\_INFO | dli | INSUPC에 LOGON할 때 해당 AS 시스템의 정보 |

<표2> Media Control Parameter

* + - 1. DB\_OPERATION\_ID

DB\_OPERATION\_ID 파라메타는 AS에서 필요한 데이터의 요청 코드이다. 사전에 정해진 ID에 따라서 DB Query 또는 기타 데이터를 수집하여 AS로 응답 송신한다.

|  |  |
| --- | --- |
| DB Server에서 미리 구현되어 있는 db 모듈 id로 AS에서 시나리오에 따라 해당 모듈 id를 수행하라고 지시한다. **정수형**의 값을 가진다.  현재는 각 서비스 마다 1 ~ 65535의 값을 갖도록 되어 있다. | |
| length | 2byte |
| Default value | N/A |
| Format  및 value | 0001 (1번 모듈 id)  0102 (258번 모듈 id) |

* + - 1. DB\_OPERATION\_NAME

DB\_OPERATION\_NAME 파라메타는 AS에서 필요한 데이터의 요청을 사전에 정의된 Procedure Name에 따라서 INSUPC의 Procedure를 수행하여 그 결과를 AS로 응답 송신한다.

|  |  |
| --- | --- |
| DB Server에서 미리 구현되어 있는 db Procedure name으로 AS에서 시나리오에 따라 해당 procedure를 수행하라고 지시한다..  INSUPC에서는 Operation Name + Operation ID 필드를 갖는 Mapping Table을 관리하여 내부적으로는 Operation ID값을 기준으로 동작하도록 한다. | |
| length | 가변 (최대 40Byte) |
| Default value | N/A |
| Format  및 value | Operation name이 “123” 인 경우  03 31 32 33 (03은 operation name 자리수, 나머지 세 byte는 asciii code 값) |

* + - 1. DB\_INPUT

DB\_INPUT 파라메타는 AS에서 요청한 데이터를 처리할 때 필요한 데이터 항목을 나열한다.

|  |  |
| --- | --- |
| DB 모듈을 수행하기 위해서 필요한 input data로 주로 sql 문에서 table filed 변수 값으로 사용된다. 필요 없는 경우 생략된다. 문자형의 값을 가진다. | |
| length | 가변 |
| Default value | N/A |
| Format  및 value | String (ASCII code)  Data 길이 자릿수 : 2byte , 최대데이터길이: 1024byte  예) input parameter가 “123” “4567” “78901” 세개의 string인 경우  03 00 03 31 32 33 04 34 35 36 37 05 37 38 39 30 31  03 (input data 개수)  **0003** (첫번재 input data 자리수 2byte) 31 32 33 (“123”)  0004 (두번째 input data 자리수 2byte) 34 35 36 37(“4567”)  0005 (세번째 input data 자리수 2byte) 37 38 39 30 31(“78901”) |

* + - 1. DB\_OUTPUT

DB\_OUTPUT 파라메타는 AS에서 요청에 따라 수집된 결과데이터 항목을 나열한다.

|  |  |
| --- | --- |
| DB 모듈을 수행한 후 얻어지는 output data로 주로 table filed에 저장되어 있는 값으로 추후 시나리오 분기용이나 CDR data 저장용으로 사용된다. 필요 없는 경우 생략된다.문자형의 값을 가진다. | |
| length | 가변 |
| Default value | N/A |
| Format  및 value | Ex) 두개의 Record, 첫번째 record에는 2개의 필드가 존재하며 값은 각각 “123”, “1234”, 두번째 record에는 1개의 필드가 존재하며 값은 “123”일 경우아래와 같은 데이터가 output parameter로 전송된다.  02 02 00 03 31 32 33 00 04 31 32 33 34 01 00 03 31 32 33  레코드 수 1 번째 레코드 2번째 레코드  1번째 레코드>  02 00 03 31 32 33 00 04 31 32 33 34  필드 수 1번째 필드 2번째 필드  1번째 필드>  00 03 31 32 33  필드 value(“123”)  필드 자리수 |

* + - 1. DB\_RESULT

AS에서 요청된 패킷에 대한 처리 결과를 전송한다.

|  |  |
| --- | --- |
| DB 모듈 수행 처리 결과 값으로 정수형의 값을 가진다. | |
| length | 2byte |
| Default value | N/A |
| Format  및 value | Result Category(1byte) + Result Value(1byte)   |  |  | | --- | --- | | **Result category** | **Result value** | | 0x00(Success) | 0x00 (SUCCESS)  0x01 (NO DATA) | | 0x10(AS fail) | 0x10 (NO OP ID)  0x11 (INVALID OPID)  0x12 (NO INPUT PARM)  0x13 (INVALID INPUT PARM)  0x14 (NO OP NAME)  0x15 (INVALID OP NAME) | | 0x20(DB Fail) | 0x20 (SQL operation error)  0x21 (DBMS not connected state)  0x22 (DBMS not accessible state | |
|  |  |

* + - 1. DB\_STATUS

INSUPC에서는 AS에 현재 자기 시스템의 상태를 전송한다.

최초 연결 후 로그인할 때 응답패킷에 파라메타로 전송되고, 필요에 따라(Block, …등) AS에 전송한다.

|  |  |
| --- | --- |
| INSUPC 상태정보에 대한 파라메타로 두가지의 경우에 송수신 된다.   1. 로그인할 때 INSUPC의 현재상태 전송한다. 2. 운영자에 의해 시스템 상태가 바뀔 때 상태를 전송한다. | |
| length | 2byte |
| Default value | N/A |
| Format  및 value | XX YY (2byte)  XX : 시스템 category  0x4d 🡪 ‘M’ ( main DB system)  0x44 🡪 ‘D’ (DR DB system)  YY : 시스템 status  0x41 -> ‘A’ (Active)  0x53 -> ’S’ (Standby)  0x42 -> ‘B’ (Block) |

* + - 1. DB\_LOGON\_INFO

|  |  |
| --- | --- |
| AS 시스템의 정보 | |
| Length | 4byte |
| Default value | N/A |
| Format  및 value | VCA(1byte) – 0xF0 로 fix  AS ID(1byte) (각사 납품된 순서 번호 1부터 MAX 장비까지)  Module Name ID(1byte)(각사별 프로세스 Name ID)  Net Connect ID(1byte)(각사별 AS별 TCP Connection ID ( 1부터 시작하여 최대 5까지 )   * Net Connect ID 가 현재 연결되어 있는 상태에서 재연결 요청일 경우 이전에 연결된 것을 종료하고 신규연결된 것을 사용한다. |

* + 1. Database Control Message 와 Parameter 와의 대응관계

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Message Symbol parameter** | | **dbq** | **dbp** | **dbaq** | **dbap** | **dbnq** | **dbnp** | **dbsq** | **dbsp** |
| 0x01 | DB\_OPERATIONID | O | O |  |  |  |  |  |  |
| 0x02 | DB\_OPERATION\_NAME | M | M |  |  |  |  |  |  |
| 0x03 | SQL\_INPUT | O |  |  |  |  |  |  |  |
| 0x04 | SQL\_OUTPUT |  | O |  |  |  |  |  |  |
| 0x05 | SQL\_RESULT |  | M |  |  |  |  |  |  |
| 0x06 | DB\_STATUS |  |  |  | M |  | M |  | M |
| 0x07 | DB\_LOGON\_INFO |  |  | M | M |  |  |  |  |

O : 선택(Optional) 파라미터

M: 필수(Mendatory) 파라미터

|  |  |
| --- | --- |
| dbq | DB Query 수행 요청 |
| dbp | DB Query 수행 요청 결과 |
| dbaq | DB Access 인증 요청 |
| dbap | DB Access 인증 요청 결과 |
| dbnq | AS DB 간 HEARTBEAT Request |
| dbnp | AS DB 간 HEARTBEAT Response |
| dbsq | DB 상태정보를 AS에 전송 |
| dbsp | DB 상태정보 전송에 대한 응답 |

* 1. 전문 구조 (버전 2.0)

INSUPC에서 기존 버전과 버전 2.0 프로토콜은 함께 제공된다.

* + 1. 기본 구조

|  |  |  |  |
| --- | --- | --- | --- |
| **H**  **E**  **A**  **D**  **E**  **R** | 프로토콜 버전 (2byte) | 🡺 | 0xFFFF |
| MSG\_LEN (2byte) | 🡺 | Packet Body Size, 최대 61kbyte |
| **B**  **O**  **D**  **Y** | JSON |  |  |

[그림 4-1] Packet 구조 (버전 2.0)

* + 1. 시스템 인증 규격
* 요청

{"Op":"AccReq","Seq":"2-14-180724154931-127444210",”Sid”:1,”Nid”:1}

* Sid : 시스템 ID
* Nid : 세션 ID
* 응답

{"Op":"AccRsp","Seq":"2-14-180724154931-127444210","State":"A","Result":0}

* State : MA, MB, DA, DB
* Result : 0:성공, -1:인증실패
  + 1. Heartbeat 규격
* 요청

{"Op":"HbtReq","Seq":"2-11-180724154923-764738764",”Sid”:1,”Nid”:1,}

* 응답

{"Op":"HbtRsp","Seq":"2-11-180724154923-764738764","Division":"1","State":"A","Result":0}

* Division : 세션별 분배율
  + 1. Query 규격
* 요청

{"Op":"QryReq","Seq":"3637909781156-415501137","Svc":"OT",”Api”:”cdbiGetSnSvcInfo”,"Values":["Value#1", "Value#2"]}

* Values : 서비스별 수집값 배열
* 응답

{"Op":"QryRsp","Seq":"3637909781156-415501137","Svc":"OT","Result":0}

1. INSUPC와 AS 간 HEARTBEAT 구조

* AS 와 INSUPC간 주기적으로(Config) HEARTBEAT 패킷 송수신한다.
* HEARTBEAT 은 AS->INSUPC->AS 형태로 이루어지며 AS에서 요청하고 INSUPC에서 응답하는 형식으로 진행된다.
* HEARTBEAT구조는 아래의 정상적인 경우와 비정상적인 경우의 규칙에 의해 처리된다.
* INSUPC는 HEARTBEAT 응답 전송시 시스템 상태를 전송한다.
  1. 정상적인 HEARTBEAT 송수신

AS

**요청 주기 : 10초(config)**

**Timeout : 30초(config)**

**HEARTBEAT Response**

INSUPC

**HEARTBEAT Request**

* 1. 비정상적인 HEARTBEAT(No Response)

INSUPC

AS

**HEARTBEAT Request**

**Timeout : 5초(config)**

**No Response**

1. Timeout 발생할 경우 Socket Disconnection
2. 주기적으로 Reconnection 시도
   1. 비정상적인 HEARTBEAT(No Request)

INSUPC

AS

**HEARTBEAT No Request**

**요청 주기 : 15초(config)**

1. Socket Close
2. Listen 상태 유지

이 경우 HEARTBEAT 이외에 데이터 요구가 있다면 HEARTBEAT 이 오지 않더라도 Socket 유지한다

1. INSUPC 이중화 정책
   1. 이중화 연결 구조
2. INSUPC는 Active-Active 구조를 갖는다.
3. INSUPC들은 정상의 경우 모두 서비스가 가능한 상태로 Loading되어 있어야 한다
4. AS는 등록된 INSUPC들에 대해서 모두 연결된 상태를 유지한다.
5. 최초 연결할 때 Logon하도록 하며 Logon응답을 받을 때 INSUPC로부터 DB Status를 받는다.
6. HEARTBEAT 응답 시 INSUPC로부터 DB Status를 받는다
7. INSUPC들은 자신의 상태가 변경될 경우 AS들에게 자신의 상태를 알려준다.
   1. INSUPC 이중화 정책 : Active-Active 구조

Active

AS

Active

AS

최초 연결시

Logon 인증 및

DB 상태 송수신

Active-Active 구조

DB간 상태체크

* + 1. AS 와 INSUPC 연결 전
* INSUPC
  + INSUPC에서는 서비스를 위해 필요한 모든 프로세스가 기동되어 있어야 한다.
  + 이중화 구조를 위한 최초 환경구성이 되어 있어야 한다.(A-A)
  + As로부터 연결을 위해 해당포트가 준비되어 있어야 한다.
* AS
* As는 연결을 위한 정보들이 Config 될 수 있도록 한다.
  + 1. AS 와 INSUPC 연결시도
* 공통 DB
* INSUPC는 정해진 포트로 연결되어 오는 AS들의 세션을 관리한다.
* 하나의 AS는 최대 10개까지의 세션 연결을 허용한다. 그 이상의 연결을 시도할 경우 연결을 거부한다.
* 세션 ID 는 Logon 과정에서 주고받으며, 응답시 시스템 상태를 전송한다.
* 중복된 세션 ID가 발생할 경우 이전의 세션은 연결을 끊고 새로운 세션을 사용한다.
* 30초 이내 (HeartBeat을 3회 수신하지 못 할 경우) 서비스 요청 전문 또는 HeartBeat 수신을 하지 못 할 경우 해당 세션을 종료한다.
* AS
* AS는 INSUPC들에 대하여 모두 연결상태를 유지한다.
* 하나의 As는 최대 10개의 세션을 연결할 수 있다.
* AS는 연결이 허용된 세션으로만 패킷을 주고받을 수 있다.
* Config 된 INSUPC연결이 안될 경우 주기적으로 재연결을 시도한다.(1초주기)
  + 1. AS 와 INSUPC 패킷 송수신
* 공통 DB
* INSUPC는 As로부터 수신된 패킷을 처리하여 수신된 세션으로만 응답을 준다.
* 서비스 도중 연결이 끊어질 경우 이미 요청된 패킷에 대한 정보는 해당 세션이 끊어진 상태임으로 폐기한다.
* AS로부터 수신된 패킷 처리 시간이 일정 시간을 넘으면 별도로 count하여 통계를 제공한다.
* AS
* AS는 Active인 INSUPC 세션으로만 패킷을 주고받을 수 있다.
* 서비스 도중 세션 연결이 끊어질 경우 재연결을 시도하며, 이미 끊어진 세션으로 요청된 정보에 대해서는 재요청을 시도한다.
* Active 인 공통 DB가 2개 이상일 경우 패킷 송수신에 대하여 Load Sharing한다.
  + 1. AS 의 INSUPC 재연결
* AS
* AS는 INSUPC와의 연결이 끊어질 경우 재연결을 시도한다.
* As는 INSUPC에서 수신된 패킷에 문제가 있을 경우 연결을 끊고 재연결을 시도한다.
* AS는 INSUPC연결정보를 환경파일로 갖는다.
* AS는 세션이 단절 될 경우 Peer INSUPC로 수신 받지 못한 서비스 요청에 대해 재 전송을 한다.
* 모든 연결세션은 연결상태 체크를 위해 10초 주기로 HeartBeat을 송수신 한다.
* HEARTBEAT reponse 패킷 3회 연속 미 수신시 기존 연결 종료 후 재 연결을 시도한다
  + 1. AS 와 INSUPC 절체
* 공통 DB
* INSUPC는 상호간에 상태체크를 하여 이중화모드에 맞도록 상호 상태를 유지한다.
* Active인 INSUPC는 자신의 상태를 체크하여 서비스 처리를 할 수 없는 경우 절차에 따라자신의 상태를 BLOCK상태로 전환한다..
* AS
* AS는 INSUPC가 Active인 상태를 Block상태로 전환할 경우 패킷 전송을 중지한다.
* AS는 DB로부터 Active상태를 수신하게 되면 해당 INSUPC로 패킷을 송수신한다.
  + 1. INSUPC 장애 발생 시 정책
* AS에서 응답패킷을 수신하지 못할 경우
* AS는 요청 패킷을 송신 후 INSUPC로부터 응답패킷이 오지 않는 경우가 연속적으로 발생할 경우 일정시간 후 장애를 발생시키고 다른 Active DB로 패킷을 송수신한다.
* INSUPC에서 장애가 발생할 경우
* INSUPC에서는 내부 시스템 상태체크 시 서비스를 계속적으로 할 수 없는 Critical 장애가 발생할 경우 알람을 발생시키고 AS로 DB STATUS를 BLOCK 상태로 전송하여 패킷이 수신되지 않도록 한다.
* AS는 BLOCK 모드인 DB로 패킷을 전송하지 않는다.