1. 클래스 다이어그램 (Class Diagram)

+---------------------------------------+

| NetworkDevice |

|---------------------------------------|

| - log\_data: list |

| - network\_state: dict |

| - network\_state\_lock: threading.Lock |

| + connect\_device(host: str): ConnectHandler |

| + gather\_network\_stats(child: ConnectHandler): dict |

| + read\_cli\_and\_write\_to\_csv(m\_box: str): void |

| + check\_thresholds\_and\_trigger\_event(results: dict, dev\_ip\_addr: str, interface: str): bool |

+---------------------------------------+

+---------------------------------------+

| Logger |

|---------------------------------------|

| + write\_to\_csv(filename: str, data: dict): void |

| + save\_log\_to\_text\_file(file\_base\_name: str, event\_type: str, message: str, output: str): void |

+---------------------------------------+

+---------------------------------------+

| EventHandler |

|---------------------------------------|

| + handle\_event(event\_type: str, message: str, dev\_ip\_addr: str, interface: str): void |

| + handle\_event\_and\_predict(results: dict, m\_box: str): void |

| + predict\_link\_down\_with\_smoothing(results: dict): void |

| + process\_syslog\_message(message: str, client\_address: tuple): void |

| + execute\_cli\_command(child: ConnectHandler, commands: list, cmd\_type: str): list |

+---------------------------------------+

+---------------------------------------+

| ModelTrainer |

|---------------------------------------+

| - MIN\_DATA\_COUNT: int |

| - collected\_data: list |

| + train\_model(data: list): bool |

| + load\_training\_data(): bool |

| + collect\_and\_train(): void |

+---------------------------------------+

+---------------------------------------+

| PredictWithSmoothing |

|---------------------------------------+

| - model: RandomForestClassifier |

| - alpha: float |

| - prev\_prediction: np.ndarray |

| + predict(X: np.ndarray): np.ndarray |

+---------------------------------------+

+---------------------------------------+

| SyslogUDPHandler |

|---------------------------------------+

| + handle(): void |

+---------------------------------------+

2. 시스템 구성 요소 다이어그램 (Component Diagram)

+----------------------+ +------------------+

| NetworkDevice | | EventHandler |

+----------------------+ +------------------+

| + connect\_device() |<>----------->| + handle\_event()|

| + gather\_network\_stats()| | + predict\_link\_down\_with\_smoothing()|

| + read\_cli\_and\_write\_to\_csv()| | + process\_syslog\_message()|

| + check\_thresholds\_and\_trigger\_event()| + handle\_event\_and\_predict()|

+----------------------+ +------------------+

| |

v v

+----------------------+ +------------------+

| Logger | | ModelTrainer |

+----------------------+ +------------------+

| + write\_to\_csv() | | + train\_model() |

| + save\_log\_to\_text\_file()| | + load\_training\_data()|

+----------------------+ | + collect\_and\_train()|

+------------------+

|

v

+----------------------+

|PredictWithSmoothing |

+----------------------+

| + predict() |

+----------------------+

|

v

+----------------------+

| SyslogUDPHandler |

+----------------------+

| + handle() |

+----------------------+

3. 주요 절차에 대한 시퀀스 다이어그램 (Sequence Diagrams)

3.1 Transceiver 정보 수집 절차

Actor(Client) -> NetworkDevice : connect\_device(host)

NetworkDevice -> ConnectHandler : ConnectHandler

ConnectHandler -> NetworkDevice : gather\_network\_stats(child)

NetworkDevice -> Logger : write\_to\_csv(filename, data)

Logger -> CSV File : append data to CSV

3.2 로그 CSV 파일로 저장 및 이벤트 처리 절차

Actor(Client) -> NetworkDevice : connect\_device(host)

NetworkDevice -> ConnectHandler : ConnectHandler

ConnectHandler -> NetworkDevice : gather\_network\_stats(child)

NetworkDevice -> Logger : write\_to\_csv(filename, data)

Logger -> CSV File : append data to CSV

NetworkDevice -> EventHandler : handle\_event(event\_type, message, dev\_ip\_addr, interface)

EventHandler -> NetworkDevice : execute\_cli\_command(child, get\_commands, "get")

EventHandler -> NetworkDevice : execute\_cli\_command(child, set\_commands, "set")

EventHandler -> Logger : save\_log\_to\_text\_file(filename, event\_type, message, output)

Logger -> Text File : save log

3.3 모델 학습 절차

Actor(Client) -> ModelTrainer : load\_training\_data()

ModelTrainer -> NetworkDevice : gather\_network\_stats(child)

NetworkDevice -> Logger : write\_to\_csv(filename, data)

Logger -> CSV File : append data to CSV

ModelTrainer -> ModelTrainer : collect\_and\_train()

ModelTrainer -> ModelTrainer : train\_model(data)

ModelTrainer -> CSV File : save training data to CSV

3.4 Syslog 이벤트 처리 절차

SyslogUDPHandler -> NetworkDevice : process\_syslog\_message(message, client\_address)

NetworkDevice -> EventHandler : handle\_event(event\_type, message, dev\_ip\_addr, interface)

EventHandler -> NetworkDevice : execute\_cli\_command(child, get\_commands, "get")

EventHandler -> NetworkDevice : execute\_cli\_command(child, set\_commands, "set")

EventHandler -> Logger : save\_log\_to\_text\_file(file\_base\_name, event\_type, message, output)

Logger -> Text File : save log

4. 통합 클래스 다이어그램

+---------------------------------------+

| NetworkDevice |

|---------------------------------------|

| - log\_data: list |

| - network\_state: dict |

| - network\_state\_lock: threading.Lock |

| + connect\_device(host: str): ConnectHandler |

| + gather\_network\_stats(child: ConnectHandler): dict |

| + read\_cli\_and\_write\_to\_csv(m\_box: str): void |

| + check\_thresholds\_and\_trigger\_event(results: dict, dev\_ip\_addr: str, interface: str): bool |

+---------------------------------------+

|

v

+---------------------------------------+

| Logger |

|---------------------------------------|

| + write\_to\_csv(filename: str, data: dict): void |

| + save\_log\_to\_text\_file(file\_base\_name: str, event\_type: str, message: str, output: str): void |

+---------------------------------------+

|

v

+---------------------------------------+

| EventHandler |

|---------------------------------------|

| + handle\_event(event\_type: str, message: str, dev\_ip\_addr: str, interface: str): void |

| + handle\_event\_and\_predict(results: dict, m\_box: str): void |

| + predict\_link\_down\_with\_smoothing(results: dict): void |

| + process\_syslog\_message(message: str, client\_address: tuple): void |

| + execute\_cli\_command(child: ConnectHandler, commands: list, cmd\_type: str): list |

+---------------------------------------+

|

v

+---------------------------------------+

| ModelTrainer |

|---------------------------------------+

| - MIN\_DATA\_COUNT: int |

| - collected\_data: list |

| + train\_model(data: list): bool |

| + load\_training\_data(): bool |

| + collect\_and\_train(): void |

+---------------------------------------+

|

v

+---------------------------------------+

| PredictWithSmoothing |

|---------------------------------------+

| - model: RandomForestClassifier |

| - alpha: float |

| - prev\_prediction: np.ndarray |

| + predict(X: np.ndarray): np.ndarray |

+---------------------------------------+

|

v

+---------------------------------------+

| SyslogUDPHandler |

|---------------------------------------+

| + handle(): void |

+---------------------------------------+