

OpenWRT 기반의 무선 네트워크

통합 관제 시스템



..

부산대학교 정보컴퓨터공학부

지도교수: 김태운

팀 명: 와신상담

팀 원: **201924657** 장원석

202155508 권내현

202155547 남원정

〈목차〉

1. 프로젝트 개요.....	3
1.1 배경 및 필요성.....	3
1.2 과제 목표 및 기대효과.....	3
 2. 요구사항 분석과 제약 사항.....	 4
2.1 요구사항 분석.....	4
2.2 제약 사항 및 대책.....	4
 3. 프로젝트 소개 및 설계.....	 5
3.1 개발 환경 및 기술 스택.....	5
3.2 시스템 동작 방식 설명.....	6
3.3 과제 세부 기능 설명 및 개발 내용.....	7
 4. 개발 일정 계획 및 담당 업무.....	 9
4.1 개발 일정.....	9
4.2 담당 업무.....	10

1. 프로젝트 개요

1.1 배경 및 필요성

1.1.1 다중 공유기 제어의 어려움

대학, 회사 등 규모가 큰 시설에서 여러 대의 공유기를 제어 및 관리하는 서비스가 필요하다. 공유기를 제어하는 방식 중 하나로, 무선랜 라우터를 리눅스로 제어할 수 있도록 만들어진 운영체제인 OpenWRT를 설치하고 웹 GUI인 Luci를 통해 공유기를 조작하는 방식을 채택할 수 있다. 해당 방식은 안정적으로 라우터에 대한 전반적인 제어를 할 수 있지만, Luci를 통한 공유기 조작은 단일 공유기에 대해서만 동작하기 때문에 여러 대의 공유기 설정을 일괄적으로 변경하고 싶은 경우 일일이 처리하므로 작업의 효율이 떨어진다.

1.1.2 최적의 네트워크 설정 및 연결 어려움

기존의 스마트폰, 태블릿, 노트북 등의 기기들은 사용자가 가장 안정적이고 빠른 연결을 선택할 수 있도록 와이파이의 목록을 신호 세기 순으로 정렬하여 표시한다. 일반적으로 신호 세기가 강할수록 네트워크의 품질이 좋고 연결 속도도 빠를 가능성이 높기 때문이다. 그러나 신호 세기가 강한 공유기라 하더라도 많은 사용자가 접속해 있을 경우, 병목 현상이 발생하여 원활한 통신을 하지 못하는 일이 잦다. 특히 축제 등 사용자가 밀집된 공간에서 많은 사람들이 하나의 라우터에 연결되어 있는 경우, 연결되어 있는 모든 사용자가 네트워크를 사용할 수 없게 되기 때문에 불편함을 겪는다.

1.2 과제 목표 및 기대효과

본 과제는 OpenWRT를 이용하여 다중 공유기를 모니터링 및 제어할 수 있는 시스템을 개발하는 것을 목표로 한다. 네트워크 관리자는 앱에 로그인하면 다중 공유기 모니터링 및 제어 서비스를 통해 공유기 모니터링 및 관리의 효율을 올릴 수 있다. 또한, 다양한 상황에 최적화될 수

있도록 미리 정의해둔 네트워크 설정을 사용함으로써 병목현상을 해결하여 네트워크 연결 품질을 개선하거나, 최대한 많은 사용자에게 일정한 네트워크 연결 품질을 제공하도록 할 수 있다.

신호 세기만 고려하는 시스템의 연결 가능 공유기 리스트와 달리, 현재 연결할 수 있는 공유기 중에서 신호 세기, 작업량, 트래픽 등의 다양한 메트릭을 고려하여 최적의 공유기에 접속하는 알고리즘을 개발하여 네트워크 연결의 가용성을 높인다.

2. 요구사항 분석과 제약 사항

2.1 요구사항 분석

2.1.1 사용자 인증

- 사용자의 아이디, 비밀번호, 역할 등을 저장한다.
- 서버에 사용자의 데이터를 저장 및 관리한다.

2.1.2 다중 공유기 모니터링

- 공유기의 메트릭 정보를 수집하여 서버에 저장한다.
- 관리자는 여러 대의 공유기를 한 번에 모니터링한다.
- 관리자는 문제가 발생한 공유기의 종류를 알림을 통해 확인한다.

2.1.3 다중 공유기 제어

- 관리자는 다양한 시나리오에 따라 여러 공유기의 설정을 한번에 변경한다.

2.1.4 서비스 제공

- 사용자는 최적의 네트워크 연결을 제공하는 공유기에 연결한다.
- 관리자는 안드로이드 앱을 통해 모니터링 및 제어 서비스를 이용한다..
- 사용자 역할(관리자/일반 사용자)에 따라 접근 권한을 부여하여 보안성을 높인다.

2.2 제약 사항 및 대책

2.2.1 제약 사항

- 각 공유기 별로 메트릭을 수집/저장할 때 동작 주체가 공유기일 경우, 오버헤드 양이 크다.
- 원격으로 공유기에 접속하려면 공유기가 WAN에 연결되어 있어야 한다.
- 공유기가 갑자기 강제종료 되는 경우 문제가 발생했는지 확인하기 어렵다.

2.2.1 대책

- 공유기가 주기적으로 서버에 메트릭 값을 보내주는 방식 대신 서버가 직접 공유기 내부 OpenWrt에 접속하여 메트릭 값을 읽어 오는 방식을 채택한다.
- 주기적으로 공유기의 상태를 점검하여 문제가 발생한 경우 알림을 보낸다.

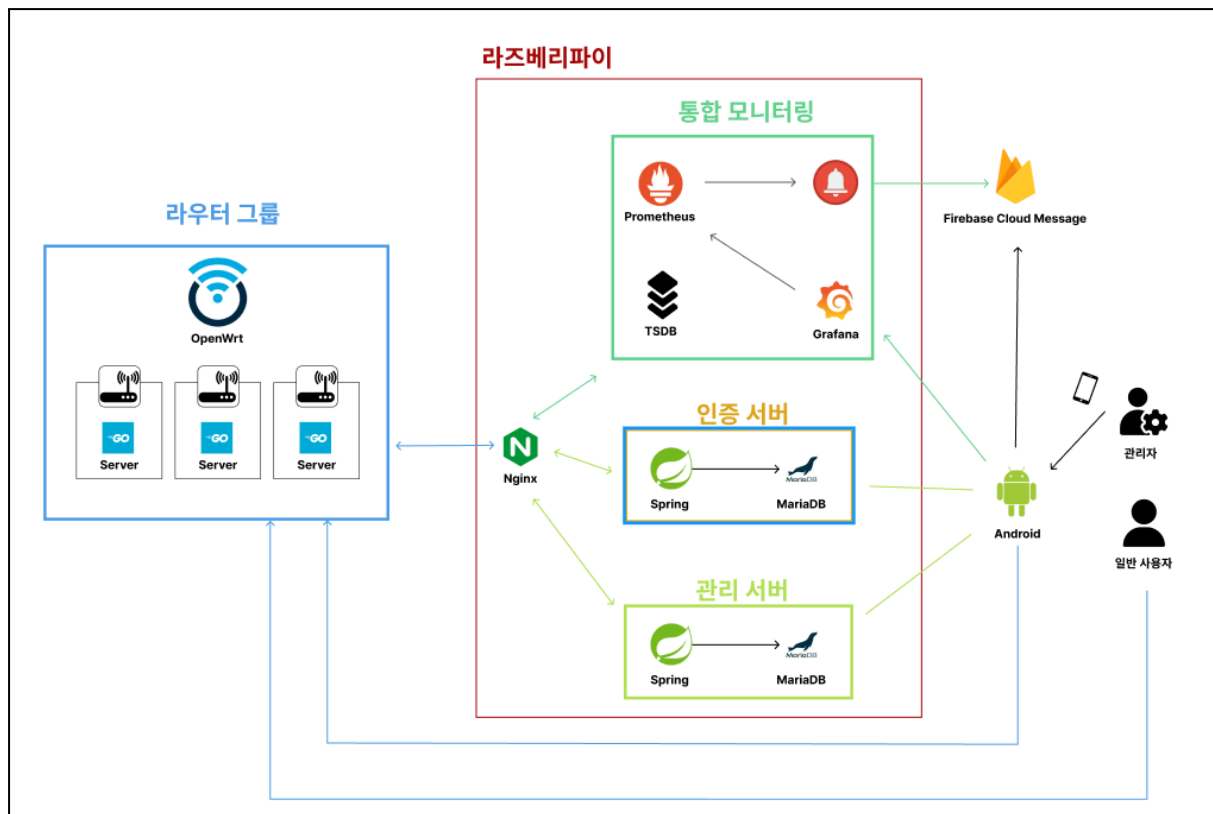
3. 프로젝트 소개 및 설계

3.1 개발 환경 및 기술 스택

사용처	기술 스택	단위	활용
라우터	OpenWRT	플랫폼	라우터 용 경량 Linux
	Golang	인프라	라우터 메트릭 전달 및 상태 제어
	Luci	소프트웨어	라우터 메트릭 수집
서버	Nginx	백엔드	
	Spring Boot	백엔드	사용자 권한 관리 및 프론트엔드 요청 처리
	MySQL	인프라	사용자 정보 및 라우터 제어 데이터 저장
	Prometheus	모니터링	라우터 메트릭 수집 및 시계열 데이터 저장
	Grafana	모니터링	라우터 메트릭 시각화 및 모니터링
	Docker	백엔드	서버 기술스택 컨테이너화 및 배포
	Docker Compose	백엔드	컨테이너 이미지 간 종속성 설정 및 네트워크/볼륨 등 세부설정 명시
인프라	Git	형상 관리	개발 협력 및 형상 관리

	Github Action	CI/CD	테스트 자동화 및 배포 자동화
	Firebase	인프라	서버-모바일 기기 간 메시지큐 / PUSH 알림
모바일 기기	Android	프론트엔드	유저 진입점 및 라우터 Hand-off 제공

3.2 시스템 동작 방식 설명



〈 그림 1. 사용자, 라우터 그룹, 관리 서버 간 다이어그램 〉

3.2.1. 멀티 라우터 지표 수집

OpenWRT 기반 라우터에서 실행 중인 **Go언어** 서버가 평상시 UNIX 관리툴을 통해 라우터 상태에 관한 주요 지표를 메모리에 수집한다. 이 서버는 주기적으로 중앙 서버 측 **Prometheus** 모니터링 시스템에 지표를 Push한다. Prometheus는 이를 **시계열 데이터** 형식으로 파싱하고, 전용 데이터베이스에 저장한다.

3.2.2. 멀티 라우터 관리

사용자는 **안드로이드** 기기로 와이파이에 연결된 라우터를 통해 통합 모니터링 서버에 접속한다. 네트워크 관리자의 접근만 허용하기 위해 **인증 서버**를 거쳐 토큰을 발급받아 서버에 접근하며, **Grafana** 웹 인터페이스를 통해 관리 권한이 있는 라우터의 주요 시계열 데이터를 시각적으로 확인한다. 관리자는 여기서 필요에 따라 알림 조건을 설정할 수 있다.

3.2.3. 라우터 프로파일 설정

관리자는 **인증 서버**로부터 발급받은 토큰을 담아 Java **Spring** 기반의 **관리 서버**에 접근한다. 관리 서버는 프록시로 동작하는 **Nginx**를 거쳐 간접적으로 라우터 그룹과 연결된다. 관리자는 라우터의 상태와 관련한 특정 조건과 조건을 만족할 시 실행할 일련의 절차(Action)를 담은 **프로파일**을 준비하고, 관리 서버에 등록한다. 관리 서버는 프로파일을 파싱하여 **통합 모니터링 서버**에 알림 조건을 등록 또는 수정하며, 프로파일 조건이 만족되면 라우터 그룹에 Action을 취한다.

3.3 과제 세부 기능 설명 및 개발 내용

1) OpenWrt 및 Luci 설치를 통한 기본 환경 구축

여러 대의 공유기를 효율적으로 제어하기 위해 공유기에 OpenWrt를 설치하여 사용한다. 이 때, 환경 구축을 위해 총 4개의 공유기를 사용할 것이며, OpenWrt를 설치할 모델로는 IPTIME-AX2004m 모델을 이용할 것이다. IPTIME-AX2004m은 중급 성능을 가지는 5G 공유기 중 OpenWrt를 설치하는 과정이 가장 간편하기 때문에 이번 과제에서 주로 사용할 예정이다.

이후, 공유기를 WAN에 연결한 후 Luci를 통해 공유기의 방화벽을 설정하여 외부에서도 원격으로 OpenWrt 또는 Luci에 접속할 수 있도록 설정한다. 원격으로 해당 공유기에 접속할 수 있음을 확인하면 모니터링 및 관리를 하기 위해 필요한 공유기의 Metric을 확인한다. 이번 과제에서는 연결된 네트워크 장치의 수, 수신/전송된 패킷의 수, 데이터의 양, 수신/전송 중 error 발생 빈도 등의 정보를 수집할 계획이다. 해당 데이터들은 Luci의 프로토콜로 정의되어 있기 때문에 주기적으로 수집한 정보들을 정리하여 한 파일에 기록한

후, 원격 네트워크에서 접속한 서버가 해당 데이터를 가져가는 방식으로 진행된다. 주기적으로 공유기의 데이터를 수집하기 위해서 javascript를 이용한 프로그램을 사용할 계획이며, 그렇게 수집한 데이터는 각 공유기 별로 구별되어 DB에 저장된다.

2) Prometheus, Grafana를 이용한 다중 공유기 모니터링 시스템 구축

데이터베이스에 저장된 각 공유기의 Metric 데이터를 이용하여 여러 대의 공유기의 상태를 확인할 수 있는 시스템을 구축한다. 모니터링 시스템을 구축하는 데에는 Prometheus, Grafana를 사용할 계획이며, 개별 공유기 모니터링과 통합 공유기 모니터링을 모두 지원한다. 모니터링 서비스는 웹을 이용해 접속할 수 있다. 해당 공유기들의 관리자만이 모니터링 서비스에 접근할 수 있으며, 사용자 권한은 로그인 서비스를 통해 관리한다.

측정한 Metric이 정상 범주를 벗어나거나 문제가 생겨 Metric을 보낼 수 없는 공유기가 있을 경우, 관리자에게 문제가 생긴 공유기의 SSID와 발생한 문제를 푸시 알림을 통해 안드로이드에서 확인할 수 있도록 설계한다. 이 오류는 모니터링 시스템에서도 확인할 수 있다.

3) 공유기 통합 관리 서비스 개발

단일 공유기를 제어하기 위해서는 Luci를 이용하여 공유기를 제어하는 기존의 방식을 사용할 수 있다. 관리자는 원격으로 접속할 수 있는 DNS 주소를 이용해 Luci에 접속할 수 있으며, 관리자가 아닌 사용자는 Luci를 통한 접속이 제한된다. 모든 공유기에 같은 설정을 적용하고 싶을 때에는 특정 상황에 가장 적합한 성능을 발휘할 수 있는 ‘프로필’을 사용할 수 있다. 프로필을 통해 사용자의 네트워크 연결 품질을 개선하거나 공유기를 효율적으로 운영할 수 있다. 관리자가 특정 프로필을 선택하면 모든 공유기에 해당 프로필 설정이 적용되는 방식으로 동작하며, 사용할 수 있는 프로필 목록은 다음과 같다.

- 공유기들을 평소와 같은 방식으로 운영한다. 이 상태를 default로 지정한다.
- 많은 사람들이 공유기에 접속해 있을 경우, 최대한 많은 사람들이 네트워크를 원활하게 이용할 수 있도록 하기 위해 네트워크에 접속해 있는 상태에서 일정 시간 동안 트래픽 교환이 발생하지 않은 기기를 자동으로 접속 해제한다. 또한, 네트워크에 사람이 너무 많이

물리는 것을 사전에 방지하기 위해 ‘이용자 대기열’ 시스템을 도입한다. 이용자 대기열은 일종의 큐와 같은 역할을 하며, 접속을 요청한 순서대로 대기를 하는 방식으로 작동한다.

- 공유기를 이용하는 사람이 적어 아무도 접속해 있지 않은 공유기가 많이 존재할 경우, 공유기들을 절약 모드로 운영한다.

4) 내부 네트워크에서 자체적인 hand-off 기능 개발

hand-off란 공유기 사용자가 물리적으로 이동하면서 기존에 접속 중이던 공유기의 신호 세기가 약해짐에 따라 기존의 공유기에서 다른 공유기로 접속을 전환하는 과정을 의미한다. hand-off는 일반적으로 무선 네트워크에서 끊김 없는 연결을 유지하기 위해 중요하다. 신호 세기를 중심으로 확인하던 기존 hand-off 기능과 달리 이 시스템에서는 신호 세기 및 대역폭, 데이터 전송 중 오류 발생 빈도 등을 통합적으로 고려하여 hand-off를 발생시킬 것이다. 신호 세기, 대역폭, 오류 발생 빈도 등은 각 공유기에서 수집한 Metric을 사용한다.

Metric을 기반으로 현재 접속할 수 있는 최적의 공유기를 찾는 알고리즘은 공유기에 접속해 있는 단말에서 직접 계산한다. 최적의 공유기를 찾으면 그 공유기에 자동으로 연결되거나, 또는 사용자가 연결 여부를 직접 결정할 수 있다.

실제 서비스는 안드로이드를 통해 이용할 수 있도록 개발할 계획이다. 서비스 이용자는 관리자와 관리자가 아닌 자로 구별할 수 있는데, 관리자가 아닌 사용자는 hand-over 기능만 사용할 수 있다. 공유기에 직접 접근하거나 설정을 변경할 권한이 없다. 반면에 관리자는 hand-over기능 뿐만 아니라 공유기 모니터링 서비스와 통합 관리 시스템을 사용할 수 있다.

4. 개발 일정 계획 및 담당 업무

4.1 개발 일정

구분	작업 일정																	
	5월			6월				7월				8월					9월	
	3	4	5	1	2	3	4	1	2	3	4	1	2	3	4	5	1	2

환경 및 분석																		
요구사항 분석																		
시스템 환경 구축																		
사용자 권한 관리 API개발																		
공유기 데이터 수집 및 저장																		
UI 개발 및 API 연동																		
다중 공유기 모니터링 시스템 구축																		
라우터 문제 발생 알림 기능 구현																		
상황에 따른 공유기 설정 변경																		
앱 내 hand-off																		
시스템 테스트 및 디버깅																		
배포 테스트 및 배포																		
최종 보고서 및 발표 준비																		

4.2 담당 업무

이름	분류	세부 역할 분담
----	----	----------

장원석	인프라	<ul style="list-style-type: none"> • 구축 <ul style="list-style-type: none"> - 라즈베리파이 환경 구축 - Docker 컨테이너화 - Nginx 구축 - Prometheus, Grafana 구성 - Github Action 구성
	모니터링	<ul style="list-style-type: none"> • Prometheus로 지표 데이터 수집 • Grafana 연동해서 대시보드 설정
권내현	플랫폼	<ul style="list-style-type: none"> • OpenWrt - Luci 환경 구축 <ul style="list-style-type: none"> - OpenWrt를 기반으로 한 공유기 환경 구축 및 WAN 연결 - SSH를 이용한 공유기 원격 접속 기능 설정 - 각 공유기에 대한 DNS 주소 설정 • 원격 서버로 Metric 수집 <ul style="list-style-type: none"> - Luci 프로토콜을 이용해 Metric 수집 - 수집한 Metric을 원격 서버에 저장 - 수집한 데이터를 각 공유기 별로 저장하는 DB 구축
	백엔드	<ul style="list-style-type: none"> • 공유기 통합 관리 서비스 구현 <ul style="list-style-type: none"> - Luci 프로토콜을 이용하여 공유기 제어 - 각 상황에 대한 프로필 설정 • 공유기 보안 설정 <ul style="list-style-type: none"> - 방화벽을 이용해 관리자 외의 사용자가 SSH를 통해 OpenWrt에 접속하는 것을 방지 - 관리자 외의 Luci에 대한 원격 접속, 로컬 접속을 막음
남원정	프론트엔드	<ul style="list-style-type: none"> • UI 개발 <ul style="list-style-type: none"> - Android 를 사용해서 로그인, 회원가입 기능이 있는 앱 제작 - 다중 공유기 모니터링 - 상황에 따라 공유기 설정 변경 - 가장 최적의 네트워크 연결을 제공하는 공유기에 연결 (hand-off)

	백엔드	<ul style="list-style-type: none"> • 사용자 인증 모듈 구현 <ul style="list-style-type: none"> - 회원가입, 로그인, 로그아웃, 관리자 권한 부여 기능 • 공유기 데이터 저장 • 공유기 문제 발생 시 알림 전송
	DB 모델 설계	<ul style="list-style-type: none"> • 데이터 모델링 <ul style="list-style-type: none"> - 개념적, 논리적 데이터 모델 작성 - 정규화 수행 • 테이블 설계
공통	보고서 작성	• 중간 보고서, 최종 보고서
	발표 및 시연	• 발표자, 시연자, Q&A 담당자