

## 미래 전력망 안정성 분석을 위한 레거시 전력 분석 시스템과의 상호 운용성 구현

박 태 정\*

\*덕성여자대학교 사이버보안/IT미디어공학과 부교수

### Implementing Interoperability with Legacy Power Analysis Systems to Analyze Stability of Future Power Grids

Taejung Park\*

\*Associate Professor, Department of Cyber Security/Information Technology and Media Engineering, College of Engineering, Duksung Women's University, Seoul 01369, Korea

#### [요 약]

4차산업혁명을 대비하여 산업 전반에서 인공지능, 빅데이터 등 다양한 IT 기술이 적용됨에 따라 IT 분야와 다른 여러 기술 분야의 융합과 협력이 증가하고 있다. 이러한 융합을 실현하기 위해서는 다양한 산업 분야에서 소비와 생산이 결합된 프로슈머를 위한 새로운 디지털 콘텐츠 제공을 지원하는 시스템의 개발이 필수적이다. 그러나 융합 연구 개발 활동에서 서로 다른 전문 분야들 사이의 전공 지식 격차가 융합 기술 개발 진행의 큰 장애물로 작용하는 것이 사실이며 IT 관련 지식이 상대적으로 부족한 타 기술 분야 전공자들의 연구 개발과 각 산업 분야에서 최신 IT 기술의 도입을 저해하는 레거시 기술들을 포괄하여 운영이 가능한 방안을 지원하고 최신 IT 기술을 다양한 분야에 적용할 수 있도록 하는 현실적인 방법을 마련해야 한다. 본 논문에서는 이러한 실제 개발 사례에서 연구 개발된 결과로서, 중앙집중화된 데이터베이스를 중심으로 하는 레거시 전력 정보 시스템의 한계를 극복하고 미래 환경에 대응하기 위해 Python Remote Object를 이용한 인터페이스, 모듈화, 스테이트리스(stateless) 통신 등을 특징으로 하는 차세대 전력 시스템 설계 및 실체를 논의한다. 본 논문에서 소개하는 IT/전력 융합 분야에서의 실제 개발 사례와 경험은 앞으로 IT 분야 전공자들이 비 IT 분야 전문가들과 개발 과정에서 공동작업 수행이 필수적인, 다양한 산업 분야와의 융합을 위한 개발 실무 참고 자료로 활용할 수 있을 것으로 기대한다.

#### [Abstract]

As a preparation for the new era of the Industry 4.0, various industries adopt IT technologies; to implement the promises of the Industry 4.0, it is required to design and implement technological platforms to support the activities of the “prosumers” by providing new forms of digital contents. Unfortunately, IT experts still face challenges in cooperating with non-IT experts in other industries who lack in IT knowledge. Also, some legacy technologies in the industries can affect the interoperability with the state-of-the-art IT technologies. To overcome these challenges and difficulties, we have to prepare for practical ways to cooperate with the non-IT experts in other industries with less IT expertise and to cooperate the state-of-the-art IT technologies with legacy ones. In this paper, I introduce a real development case study to design and implement a future smart power grid management system where legacy power grid software assets should be involved. The proposed design features common interfaces implemented with Python Remote Objects, modular design, and stateless communication. I hope the details and discussions in this paper provide a useful guideline for both IT experts and non-IT experts.

**색인어** : 분산형 클라이언트/서버 시스템, 지능형 전력망, 알파그리드, Python Remote Objects, 프로슈머

**Key word** : Distributed Client/Server Systems, Smart Grids, Python Remote Objects, Prosumers

<http://dx.doi.org/10.9728/dcs.2021.22.2.357>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Received** 28 December 2020; **Revised** 12 January 2021

**Accepted** 27 January 2021

**\*Corresponding Author; Taejung Park**

**Tel:** 02-901-8339

**E-mail:** [tjpark@duksung.ac.kr](mailto:tjpark@duksung.ac.kr)

## I. 서론

최근 신재생 에너지의 도입, 전기차 보급으로 인해 전력망 안정성이 중요한 과제로 인식되고 있다. 특히 지금까지는 전력망 내에서 일방적으로 전력을 소비하던 주체였던 소비자가 미래의 전력망에서는 신재생 에너지를 이용한 전력 생산 또는 전기차 내 잔여 전기 재판매 등의 활동을 통해 생산까지 겸할 수 있는 프로슈머(prosumer)[1]로 활동할 가능성이 열림으로써 지금까지 정적인 특성이 강했던 전력망 구조가 시간에 따라 가변화되는 동적인 구조로 진화할 전망이다[2].

따라서 미래의 전력망이 보다 동적인 구조로 진화하게 되면 전력 시장에 적극적으로 개입하고자 하는 프로슈머를 위한 전력 정보 관련 디지털 콘텐츠를 효과적으로 제공하는 한편, 전력망의 안정성 사전 진단 및 대응을 위한 기술이 필수적이다. 이러한 안정성 진단 및 대응을 위해서 인공지능을 포함한 ICT 기반 연구가 광범위하게 진행되고 있다.

미래를 대비한 ICT 기반의 전력망 안정성 진단 및 대응 기술을 개발하기 위해서는 전력망 자산에 적용된 기술과의 상호 운용성이 확보되어야 하는데 기술적인 측면에서 안정성과 신뢰성을 최우선에 두는 전력망 기술의 특성 상 관련 기술 자산들은 과거부터 오랜 기간 동안 검증되어온 레거시(legacy) 기술들이 적용되어 운영되는 경우가 대부분이다. 또한 보안을 비롯한 여러 가지 이유로 인해 이러한 레거시 기술들을 일시에 최신 기술로 대체하는 것도 비현실적이고 비실용적이다.

따라서 미래 ICT 기반 전력망 운영 기술을 구현하기 위해서는 레거시 기술을 최신 기술과 함께 유연하게 적용할 수 있는 방법이 모색되어야 한다.

본 연구에서는 이러한 전력망 분석을 위한 레거시 기술 자산과 최신 기술을 공동으로 운영하기 위한 실제 개발 내용을 소개하고 그 과정에서 발생하는 기술적 고려 사항들을 논의한다.

## II. 본론

### 2-1 전력망 안정성 분석의 배경 및 구현 관련 문제

지금까지 전력망 운영 기술은 안정적인 전력 공급에 중점을 두고 발전해 왔다. 안정적인 전력 공급에 가장 높은 우선 순위가 부여되는 특성 상 새롭거나 창의적인 기술보다는 과거부터 오랜 기간 동안 검증되어 온 기술만 보수적으로 적용해 온 경향이 있다. 그러나 서론에서 논의한 대로 미래의 전력망에서의 가장 큰 특성은 신재생 에너지의 확산 및 프로슈머의 등장으로 인해 전력 공급 주체와 소비 주체가 고정되지 않고 시간의 흐름에 따라 가변적으로 변경될 수 있는 여러 동적인 특성들이 수반된다는 점이다. 이러한 동적인 전력망 환경에서는 현재의 정적인 전력망 환경보다 불안정성 발생 요인들이 훨씬 더 많이 개입되며 특정 지역에서 불안정성이 발생할 경우 짧은 시간 내에 전력망 전체로 확산되어 전체적인 시스템 붕괴(black-out)로 이어질 수도 있다.

따라서 동적 전력망의 성공적인 운영을 위해서는 IT 기술과 인공지능기술 등 최신 기술이 적용되어야 하나 서술한 대로 전력망 공급 및 관리 현장에서 사용되는 레거시 기술들을 활용하는 방향으로 추진될 수 밖에 없는 제약이 존재한다. 또한 기술 자산 뿐만 아니라 미래의 전력망 시스템 개발을 위해서는 전력 전공자 및 전문가들과 IT 전공자 및 전문가의 협력이 필수적인데 새로운 아이디어의 테스트를 위해 지리적으로 분산된 환경 하에서 현장에서 간단한 모듈들끼리의 네트워크 연동도 구현 단계에서 많은 어려움이 발생하는 것이 사실이다. 특히, 상이한 두 개 이상의 기술 분야가 융합될 경우 각 분야 간의 기술적 장벽으로 인해 타분야 전공자와의 의사소통 상의 어려움이 빈번하게 발생하는 것은 물론이고[3, 4] 지리적으로 떨어진 각 환경에서 개발된 전력 분석 및 관리 소프트웨어가 각각 다른 기술로 개발되었기 때문에 네트워크를 통한 서로 간의 통신 인터페이스의 구현에서도 많은 문제와 어려움이 발생한다. 예를 들어 이러한 분산 환경의 연동과 통신을 위해 현재 가장 보편적으로 적용할 수 있는 기술은 TCP/IP 프로토콜에 기반한 완벽한 REST(REpresentational State Transfer) API[5]를 구현하고 각각의 분산 모듈의 인터페이스를 이 토대 위에서 구성하는 것이지만 한정된 프로젝트 기간 내에 IT 비전공자인 전력 전공자들이 이러한 개념을 숙지하고 구현하기에는 현실적인 어려움과 한계가 존재한다.

본 논문에서 논의하는 구현 사례에서는 특정 기관에서 주관하는 전력 연구 과제를 위해 지리적으로 떨어진 5개 대학에서 전력 분석, 예측, 관리 등을 위한 모듈들을 각각 구성하고 이 모듈들의 상호 연동을 구현하였다. 연구 참여자들의 전공은 대부분 전력 전공자들로 구성되었고 소수 IT 전공 연구자들이 상호 운영성을 구현하였다. 각 대학 연구팀에서 개발한 전력 관련 소프트웨어 모듈은 C/C++, Matlab, Python, FORTRAN 모듈 등 다양한 언어와 환경에서 개발된 레거시 개발 도구 및 코드를 이용하고 있었고 이러한 모듈들 사이의 통신을 단기간 내에 구현해야 하는 상황이었다. 여러 시행착오 끝에 전력 전공자들의 IT 관련 지식 습득과 관련된 부담을 최소화하고 단기간 내 구현은 물론 관리와 운영 효율성을 보장하기 위해 Python Remote Object[6]를 통해 전체 상호 작용을 구현하였다.

본 논문에서는 전체 시스템의 기술적 타당성을 실증하기 위해 최초로 두 개 대학 간에 구성된 사례를 중심으로 논의를 진행한다.

### 2-2 그래프 이론 기반 전력망 안정성 분석을 위한 레거시 시스템과의 연동

#### 1) 그래프 이론 기반 전력망 안정성 분석 시스템 개요

기존 전력망 분석 기법에서는 시스템 전체를 미분방정식으로 표현하는 경계 조건 문제(boundary condition problem)으로 모델링하고 Newton-Raphson 등의 수치 해석[7]을 수행하는데 전력망 안정성이 붕괴된 극단적인 조건 하에서는 수렴성이 악화되어 해를 찾지 못하는 문제가 발생한다.

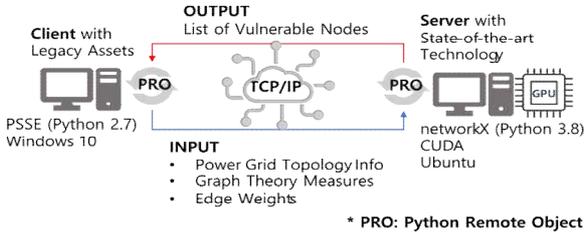


그림 1. 기술 검증 클라이언트/서버 구성도  
 Fig. 1. Structure of test client/server system for technology verification

따라서 이러한 수치 해석 기법은 일반적으로 시스템이 안정적인 정상 상태(steady state) 해석에는 적합하나 급격한 전력 사용량 변동이나 전력망 연결의 시계열 변화에 대해서는 취약한 특성을 가진다. 따라서 이 시스템의 목표는 전력망 구조를 그래프 이론에 기초하여 분석함으로써 Newton-Raphson 등 전통적으로 수치 해석 기법에 의존하는 기존 전력망 안정성 분석 방법의 단점을 극복하거나 보완하는 대안을 제시하는 것이다. 따라서 구현한 시스템에서는 수렴성 악화 문제가 발생하지 않는 그래프 이론에 기초하여 안정성이 붕괴된 전력망의 상황을 분석한다.

그래프 이론에 기초한 전력망 분석 기법은 새로운 아이디어로서 개발 과정 및 운영 시에 그 타당성을 검증하기 위해 기존 전력망 분석 시스템과의 연동을 통해 검증이 수행되어야 하는 상황이었으나 기존 전력망 분석 시스템과 그래프 분석 시스템이 동일한 컴퓨터 환경에서 구동될 수 없는 문제점이 발생하였다. 즉, 레거시 시스템에 해당되는 기존 전력망 분석 시스템(PSSE version 33 [8])은 Microsoft Windows 환경에서 2020년 이후 더 이상 지원되지 않는 Python 2.7로 인터페이스를 제공하고 있었으며 그래프 이론 기반 안정성 분석 루틴은 최신 Python 3.X 버전이 필수적인 그래프 분석 모듈(networkx [9]) 및 다중 GPU를 이용한 병렬처리를 위한 Linux 환경에서의 구현이 필수적인 상황이었다.

연구팀 인적 구성 측면에서도 기존 전력망 분석 시스템을 보유, 운영하고 있는 대학 연구팀은 전력 전공자뿐만 아니라 다중 GPU 병렬처리 서버의 운영과 관리가 어려운 실정였고 GPU 병렬처리 서버 및 그래프 이론 기반 전력망 안정성 분석 모듈을 제작하는 대학 연구팀의 경우 IT 전공자뿐만 아니라 전력망 분석 시스템의 운영과 관리가 실질적으로 불가능한 상황이었다. 이러한 기술적, 인적 구성의 한계에 따라 두 시스템을 각 대학 시설에서 분산 설치, 운영하면서 연동하는 클라이언트/서버 시스템을 설계하였다.

2) 클라이언트/서버 시스템 구성

그림 1에서는 기술 검증 시스템의 구성을 제시한다. 클라이언트에는 레거시 소프트웨어 자산에 해당되는 전력 조류(power flow) 계산 소프트웨어인 PSSE가 Windows 10 OS 환경에서 되는데 운영 중인 버전의 PSSE는 2020년 이후 더 이상 공식적으로 지원되지 않는 Python 2.7 버전으로 외부 인터페이스를 제공한다.

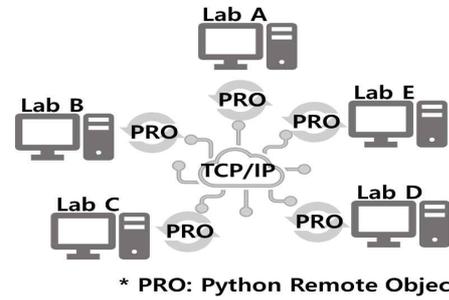


그림 2. 차세대 전력망 운영 시스템 전체 개요  
 Fig. 2. Overview of future power grid management system

서버는 병렬처리용 GPU가 4개 설치된 Ubuntu 환경으로 Python에서 그래프 이론 기반 알고리즘들이 지원되는 networkx 기반 전력망 분석 소프트웨어를 실행한다. GPU는 그래프 이론 기반 분석 루틴 중 특별히 계산량이 높은 연산(betweenness centrality [10])을 병렬로 처리하는 작업을 수행한다. networkx 모듈의 경우 다른 Python 모듈들과 마찬가지로 최신 Python 실행 환경으로 자동 업그레이드를 수행하기 때문에 동일한 실행 환경에서 Python 2.7만을 지원하는 PSSE를 함께 실행할 수 없다.

소프트웨어 구현 측면에서 레거시 소프트웨어와 최신 소프트웨어 구현이 충돌하는 경우가 흔히 발생하지만 소스 코드 수준까지 액세스가 가능한 경우 대부분은 개발 언어나 환경이 다르더라도 바이너리 레벨로 컴파일 한 후 동적 라이브러리(\*.dll 또는 \*.so)로 호출함으로써 동일한 환경 내에서 연동이 가능하다.

그러나 이전 버전의 PSSE와 같은 레거시 상용 소프트웨어 패키지의 경우 소스 코드 액세스가 불가능하며 제공하는 API 구현 사양을 통해서만 접근이 가능하기 때문에 보다 유연한 방법이 필요하다. 다양한 방법이 있을 수 있지만 앞서 논의한 대로 성능 상의 고려와 함께, PSSE와 같은 전력 기술 관련 소프트웨어 패키지를 직접 운영하고 인터페이스를 구현, 관리하는 인력은 보통 소프트웨어 개발 경험이 상대적으로 적고 Python과 같은 스크립트 언어에 대한 초보적인 개발 능력을 갖춘 전력 전공자들이기 때문에 이러한 한계를 감안하여 현실적인 대안을 제공할 필요가 있었다. 더군다나 본 논문에서 직접적으로 다루는 그래프 이론 기반 전력망 안정성 분석 모듈은 전체 미래 전력망 운영, 관리, 분석 시스템의 일부에 해당되며 향후 IT 전공 연구원들의 도움 없이 각 대학 연구 기관의 전력 전공자들이 서로 원활하게 다양한 언어와 환경에서 개발된 레거시 전력 관련 소프트웨어 모듈과 데이터를 네트워크를 통해 활용할 수 있어야 하는 상황이었다.

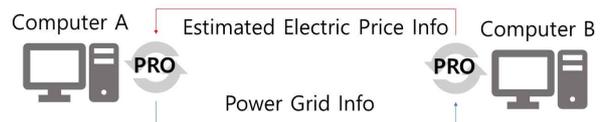


그림 3. 클라이언트/서버 단위 구성  
 Fig. 3. Unit structure for client/server

따라서 본 논문에서 소개하는 부분은 그 자체로서 전체 시스템 속에서 그래프 이론 기반 전력망 안정성 분석이라는 특정한 기능을 수행할 뿐만 아니라 전체 시스템 내 이중 레거시 모듈들 간의 협업 방식을 결정하는 테스트베드로 활용되었다.

본 논문에서는 주로 그래프 이론 기반 전력망 안정성 분석 클라이언트/서버 시스템(그림 1)의 경우를 중점적으로 다루고 있으나 최종적인 전체 시스템에서는 각 대학 연구 기관별로 전기 요금 일별 예측, 신재생 에너지 발전량 예측 및 분석, 전력 정보 시각화 등 차세대 지능형 전력망 운영 시스템을 구성하는 개별 기능을 서비스하는 서버들로 시스템이 구성되며 각각의 서버는 동시에 클라이언트로서 다른 기관의 서버에 서비스를 요청해서 의도하는 작업을 수행할 수 있도록 설계되었다(그림 2). 이 분산 시스템은 기존 설계와 다르게 레거시 시스템의 정보 자산까지 활용 가능하게 되어 전체 시스템에 대한 액세스 권한을 가지고 있는 외부 클라이언트가 전력망과 관련된 특정한 작업을 유연하게 결정할 수 있다는 장점이 있다. 예를 들어서 서버 A가 전력망 안정성 분석을 통해서 취약한 지점을 파악하고 서버 B가 프로슈머 및 재생 에너지 정보를 받고 내일 전력 생산량 예측을 제공하며 서버 C가 전력 생산 예측량을 받아서 내일 전기 요금 정보 예측을 수행한다고 가정한다면 클라이언트는 필요에 따라 단순하게 서버 B → 서버 C 순서로 시스템 불안정성을 배제한 내일 전력 생산량에 따른 전기 가격을 예측할 수도 있고 서버 B → 서버 A → 서버 C 순서로 전력 생산 예측에 따른 전력망 안정성 분석을 수행하고 불안정한 지역의 가격 상승 영향까지 고려한 결과를 얻을 수도 있다.

### 3) Python Remote Object

앞서 논의된 대로 레거시 시스템과 새로운 기술을 적용한 시스템의 상호 운영을 구현하기 위해 만족스러운 일정한 수준 이상의 시스템 작동 성능과 함께 개발 경험이 상대적으로 부족한 전력 전공자들의 소프트웨어 연동 구현 및 유지보수가 가능한 방안을 찾기 위해 몇 가지 방식을 테스트하였다.

그 결과, 두 가지 측면을 가장 적절하게 만족시키는 방식으로 Python Remote Object 기반 클라이언트/서버 환경을 구축하였다. 구축한 환경에서는 다양한 개발 언어(FORTRAN, C, Matlab, Python 2.x 버전)로 개발된 레거시 소프트웨어 모듈의 입출력 인터페이스를 Python에서 지원하는 기본 변수형 및 구조(tuple, list, dictionary 등)로 구성한다. Python 2.X 버전에서 Python 3.X 버전으로 업그레이드 되면서 여러 문법적 세부 사항들이 변경되었지만 기본적인 변수형과 구조는 그대로 유지되기 때문에 호환성 문제가 발생하지 않는다.

클라이언트/서버 각각에서 Python 인터페이스 구현은 단순히 특정 객체(object)를 생성하고 일반적인 Python 문법에 따라 그 객체가 제공하는 함수(method 혹은 member function)를 실행하도록 구현되며 네트워크를 통한 데이터 송수신은 소프트웨어 개발자에게 노출되지 않기 때문에 네트워크 개발 관련 지식이 상대적으로 부족한 전력 연구자들 입장에서는 일반적인

Python 코드로 보이며 Python 언어에 대한 기초적인 지식만으로도 지리적으로 분산된 환경 내에서도 클라이언트/서버 환경을 구축할 수 있었다.

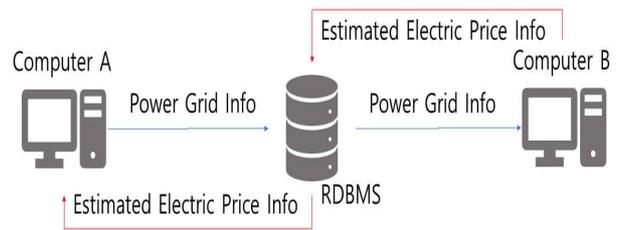


그림 4. RDBMS에 기초한 전형적인 전력망 정보 시스템 구조  
Fig. 4. Traditional Power Grid Information System Based on RDBMS

### 4) 구현 세부 사항

Python Remote Object 라이브러리로 Pyro4 [6]를 선택하였다. 서버에서는 약속된 포트 상에서 서버 daemon을 실행하고 클라이언트의 요청을 받아서 연산을 수행한 후 그 결과를 다시 전달한다. 서버에서 클라이언트로 공개할 함수(method)는 @Pyro4.expose 키워드를 이용해서 노출 여부를 지정한다. 클라이언트는 해당되는 서버의 IP 주소와 포트 번호를 이용해서 Pyro4 객체를 생성한 후 서버 측에 구현된 Python 객체의 함수(method)를 실행함으로써 의도하는 작업을 수행한다(그림 3).

클라이언트/서버 구현 시 서버에서의 다중 요청 구현을 용이하게 수행하기 위해서 RESTful API의 구현 방식 중 하나인 스테이트리스(stateless) 원칙을 적용하여 저장된 데이터를 활용하지 않도록 클라이언트에서 서버로의 요청 중 모든 정보를 서버에 제공하도록 설계하였다. 특히 전력 전공자들이 각자가 개발하는 모듈에 대해 클라이언트/서버를 구성할 때 스테이트리스 클라이언트를 적용하도록 함으로써 그 이전 중앙집중형 데이터베이스를 기초로 한 설계 시도에서 가장 큰 문제가 되었던 설계 상의 복잡성과 데이터 및 컴퓨팅 자원의 경쟁 상태(race condition)을 대부분 해소할 수 있었다. 실제로 이 프로젝트는 스테이트리스 클라이언트/서버 설계와 Python Remote Object 적용을 도입하기 전 중앙 집중형 RDBMS(Relational Database Management System)를 중심으로 각 대학 연구 기관의 전력 관련 데이터를 전달하도록 설계되었다. 이러한 방식은 메인프레임 중심의 기존 전력망 정보 시스템의 구조에 따른 것이다. 예를 들어 대학 A에서 전력망 구성 정보를 대학 B에 전달하고 대학 B에서는 이 정보를 이용하여 내일의 지역별 전기 가격을 예측한 후 다시 대학 A에 전달하려고 할 때 기존 방식에서는 대학 A가 제3의 장소에 설치, 운영 중인 RDBMS에 데이터를 저장하고 대학 B는 이 데이터를 RDBMS에서 읽어 온 후 계산하고 그 결과를 RDBMS에 저장하면 대학 A가 이 결과를 받아 오는 구조로 설계되었다. 이 구조는 연구 참여 대학들이 중간 계산 결과까지를 포함하는 전체 데이터를 공유하고자 하는 의도에서 설계되었으나 경쟁 상태가 빈번하게 발생하고 데이터 동기화 및 무결성 확보가 어려운 단점이 있었다(그림 4).

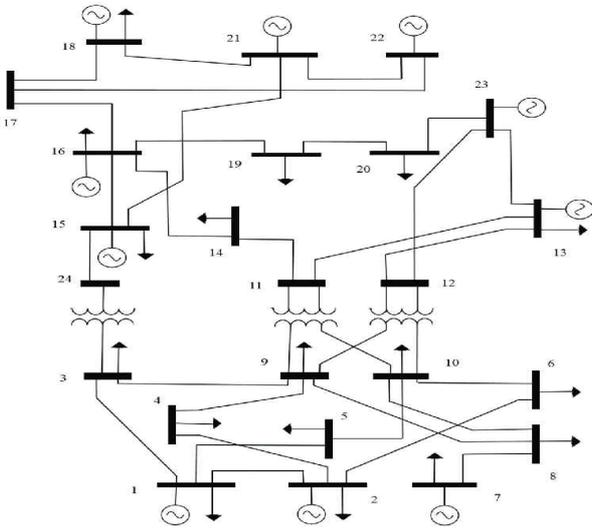


그림 5. IEEE 24 Bus 테스트 시스템  
Fig. 5. IEEE 24 Bus Test System

5) 그래프 이론 알고리즘 및 병렬화

그림 1에서 제시한 기술 검증 클라이언트/서버 구조 중 서버에서 처리하는 대부분의 연산 부하는 클라이언트로부터 받은 전력망을 그래프 구조로 구성한 후 그래프 이론에 기초한 여러 가지 측정 지표(measure)를 계산하는 작업에 집중된다. 서버는 이 결과를 조합하여 전력망 node(substation) 중 가장 취약한 지점들을 순서대로 찾아낸 후에 클라이언트로 반환한다. 그래프 이론과 전력망 안정성의 관련성에 대한 논의[11]는 이 논문의 범위를 벗어나지만 본 논문에서는 컴퓨팅 측면에서 대표적인 그래프 이론 측정 지표의 종류와 연산 복잡성의 논의를 통해 컴퓨팅 측면에서 본 논문에서 논의하는 기술 검증 구현의 세부 사항을 소개하고자 한다.

그림 5는 전력 분야에서 일반적으로 사용되는 전력망 구조를 제시한다. 이 그림에서 원 내부 물결 모양으로 표시된 항목은 발전기를 의미하고 수평 방향으로 평행한 두 개의 물결 표시는 변압기, 그리고 세로 또는 가로로 굵은 검은색 선분으로 나타낸 부분은 전력이 소비되는 지점인 버스(bus)를 의미한다.

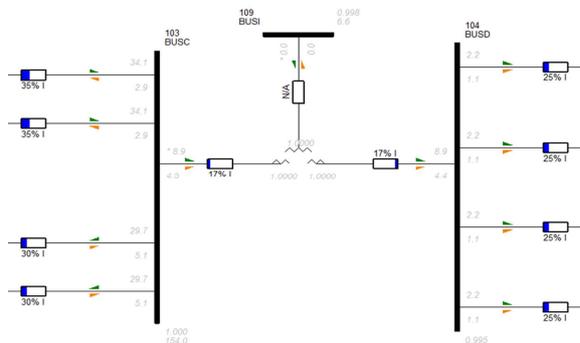


그림 6. 전력망 일부 구조  
Fig. 6. Partial Structure of Power Grid

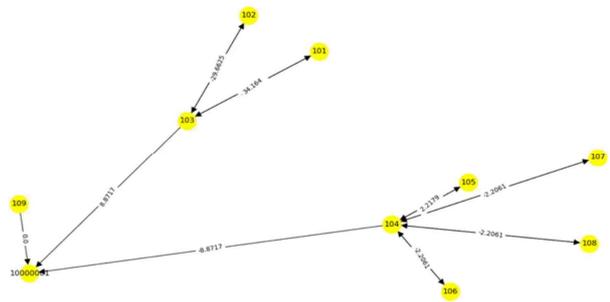


그림 7. 전력망(그림 6)의 그래프 표현  
Fig. 7. Graph Representation of Power Grid Shown in Fig. 6

그림 1에서 제시한 시스템에서는 이러한 전력망 정보를 그래프로 변환한다. 변환된 그래프의 모습은 그림 6과 그림 7에서 확인할 수 있다. 이 과정에서 그림 5에서 발전기와 변압기, 버스는 그래프의 node로, 이 요소들을 연결하는 전력선은 방향성 edge로 모델링된다. 전력선의 중요도는 edge에 특정한 가중치를 적용함으로써 지정할 수 있는데 그림 1의 시스템에서는 PSSE를 통해 계산된 각 node와 node 사이를 연결하는 전력선 상의 유효 전력, 무효 전력, 임피던스, 전력 용량을 적용하거나 단위 가중치(가중치를 1로 설정)를 선택해서 적용 가능하도록 구현하였다.

그래프 이론에 기초한 전력망 안정성 분석은 전술한 대로 본 논문이 다루는 주제를 벗어나는 범위이지만 구현 측면에서는 그래프 이론에서 널리 사용되는 그래프 특성 지표들을 조합함으로써 취약한 순위대로 node를 파악하게 된다.

이러한 목적으로 사용되는 대표적인 그래프 특성 지표는 다음과 같으며 Rubinov[10]의 정의를 따른다.

(1) Node Degree

Node degree는 가중치가 적용되지 않은 그래프의 경우 node에 연결된 edge의 개수로 정의된다. 가중치가 적용된 그래프에서는 i번째 node에 대해 다음과 같이 정의할 수 있다.

$$k_i^w = \sum_{j \in N} w_{ij} \tag{1}$$

이 때  $w_{ij}$ 는 node i와 node j를 연결하는 edge의 가중치를 의미하며 N은 node i에 연결된 주변 node들의 색인 집합을 의미한다. 전력망에서 node degree의 경우 가중치에 따라 그 의미가 달라진다. 예를 들어 유효 전력 또는 무효 전력을 가중치로 둔다면 node degree가 높은 node는 전력 소비가 상대적으로 높은 지역으로 파악할 수 있다. 또한 전력 용량으로 가중치를 둔 경우 node degree가 높은 node는 전력 설비 회사에서 이 지점의 평소 또는 잠재적인 전력 사용량이 높다고 판단하거나 절대로 과부하로 인한 전력 공급 중단이 일어나서는 안되는 지점으로 파악했을 가능성이 높다고 해석할 수 있다.

(2) Clustering Coefficient

한 node의 clustering coefficient는 해당 node가 주변 node들과의 연결이 부분적으로 끊어졌을 때 우회 경로가 얼마나 많이 확보되는가를 측정하는 지표로 볼 수 있다. 이 clustering coefficient는 방향성 그래프와 비방향성 그래프의 정의가 다르며 테스트 시스템에서는 전력망에서 특정 node가 제거 또는 붕괴되었을 때 전력 공급 흐름이 달라질 수도 있다는 가정 하에 비방향성 그래프에서의 정의를 다음과 같이 적용하였다.

$$C_i^w = \frac{2t_i^w}{k_i(k_i - 1)} \tag{2}$$

여기에서  $k_i$ 는 수식 (1)에서 정의된 i 번째 node의 node degree이며 N은 현재 node에 연결된 주변 node의 색인 집합이다.  $t_i^w$ 는 i 번째 node에 연결된 주변 node들끼리 구성하는 삼각형 개수로 정의된다. 예를 들어 그림 8에서 Node A에 대한  $t_i^w$ 는 이 node와 연결된 node들로 구성되는 삼각형이 두 개이기 때문에 2로 결정된다.

Clustering coefficient는 i 번째 node에 연결된 주변 node들끼리 연결 가능한 모든 경우의 수를 기준으로 실제로 연결된 개수의 비율을 의미한다. 따라서 수식 (2)를 다음과 같이 유도할 수도 있다.

$$C_i^w = \frac{e_i^w}{\binom{k_i}{2}} = \frac{2t_i^w}{k_i(k_i - 1)} \tag{3}$$

이 때  $\binom{k_i}{2}$ 는 조합 연산(combination)을 의미하며

$$\binom{k_i}{2} = \frac{k_i!}{2!(k_i - 2)!} = \frac{k_i(k_i - 1)}{2}$$

로 계산된다.  $e_i^w$ 는 i 번째 node와 연결된 node들 중 서로 연결된 edge의 개수로 정의되는데 이렇게 주변 node들이 서로 연결되면 결과적으로 삼각형이 형성되기 때문에  $e_i^w = t_i^w$ 이다.

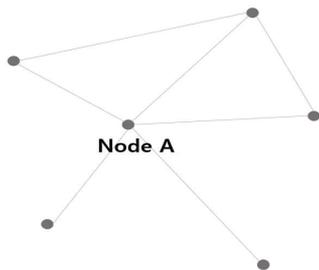


그림 8.  $t_i^w$ 의 정의

Fig. 8. The definition of  $t_i^w$

Clustering coefficient는 전력망에서 특정 node가 붕괴되었을 때 우회 대체 경로를 얼마나 많이 확보할 수 있는가를 판단하는 지표로 이해할 수 있다. 다시 말해 특정 node의 clustering coefficient가 높다면 전력 설비 회사에서 해당 지역이 중요하다고 인식해서 주변의 연결성을 강화한 결과로 해석할 수 있을 것이다. 반대로 사회 경제적으로 중요한 지점의 clustering coefficient가 낮다면 전력과 관련된 긴급한 상황 발생 시 우회 경로를 확보할 수 있도록 설비 보강을 위한 판단의 근거로 활용할 수도 있을 것이다.

(3) Betweenness Centrality

i 번째 node의 betweenness centrality는 해당 node를 제외한 그래프 내 모든 node들을 두 개씩 조합하고 두 node 사이를 최소 비용으로 이동할 수 있는 전체 최단 경로들 중에 i 번째 node를 통과하는 최단 경로의 개수를 비율로 표시한다. 두 지점 사이를 지나는 전력이 최단 비용으로 전송된다고 가정할 때 이 최단 비용을 실현하는 최단 경로에 특정 node가 포함되는 경우가 빈번하다면 해당 node는 중요성이 높은 지점이며 붕괴될 경우 전체 전력 전송 비용이 상승한다고 볼 수 있다. Betweenness centrality는 다음과 같이 정의된다.

$$b_i = \frac{1}{(n-1)(n-2)} \sum_{h,j \in N, h \neq i, h \neq j, j \neq i} \frac{\rho_{hj}(i)}{\rho_{hj}} \tag{4}$$

수식 (4)에서  $\rho_{hj}(i)$ 는 node h와 j를 지나는 최단 경로들 중 node i를 지나는 개수를 의미하고  $\rho_{hj}$ 는 node h와 j를 지나는 최단 경로들의 개수를 의미한다.

본 논문에서 논의하는 서버에서는 node degree, clustering coefficient, betweenness centrality를 networkx에서 제공하는 함수를 이용해서 구현하였다. 그림 1에서 제시한 기술 검증 시스템의 처리 속도는 레거시 시스템인 PSSE의 처리 속도, 네트워크 환경과 전력망의 크기, 연결 밀도 등에 따라 달라지지만 Python 기반 모듈인 networkx를 이용할 경우 node degree와 clustering coefficient는 edge 기준으로 약 5300개 정도의 전력망에 대해 레거시 시스템인 PSSE의 계산과 네트워크 오버헤드를 포함해도 10초 미만의 시간 내에 계산이 가능하다. 그러나 최단 거리를 계산하는 betweenness centrality는 networkx 라이브러리로 약 1시간 정도 소요되는 경우도 발생했는데 이 문제를 해결하기 위해서 GPU 가속을 통해 시간을 단축할 수 있는 옵션도 구현하였다.

6) 기존 방식의 한계 및 해결 방안

지금까지 논의한 대로 Python Remote Object를 적용함으로써 원격지 컴퓨터들 사이의 원활한 동적 클라이언트/서버 구성, IT 비전공자들의 원활한 구현 가능, 각 대학 기관의 전력 관련 레거시 자산과 최신 기술의 상호 연동 실현, 유연한 사용 시나리오 구성 등과 같은 성과를 얻었다.

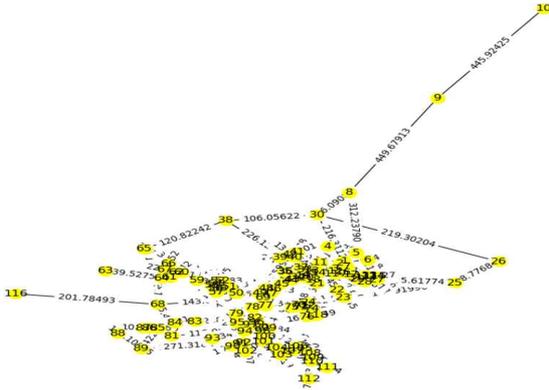


그림 9. draw\_networkx 함수 호출로 생성된 전력망 그래프의 정적 이미지(PNG)

Fig. 9. A graph image (PNG) representing a power grid created by calling draw\_networkx function

그러나 개발 과정 중에서 전력 연구를 위해 요청되는 새로운 기능들과 아이디어들을 하나 씩 추가, 개선해 가면서 예상하지 못한 문제점을 발견하게 되었는데, 전체 시스템의 인터페이스로 사용한 Pyro4 라이브러리의 경우 - 다른 Python Remote Object 모듈들과 마찬가지로 - 원격 객체 호출에 최적화된 관계로 이미지와 같은 대용량 데이터의 전송에는 적합하지 않다는 점이다. 실제로 그림 1에서 제시한 기술 시연 모듈의 개발 과정 중 전력 연구자들이 전력망 기반 전력 안정성 알고리즘 개발을 지원하기 위해 전력망 정보를 그래프 이미지로 확인하는 기능을 요청하였다. 이 기술 시연에 적용한 Python 기반 networkx 모듈은 그림 9에서 제시된 것처럼 다양한 형식으로 그래프를 정적 이미지로 출력하는 옵션을 제공한다.

최초 구현에서는 클라이언트의 요청에 의해 networkx를 실행할 수 있는 서버에서 networkx 모듈의 그래프 이미지 출력 함수(draw\_networkx 함수)를 호출하여 서버에서 그림 9와 같은 이미지를 생성한 후 이 이미지의 픽셀 RGB 값을 Python 표준 List로 변환한 후 클라이언트로 전송하고 클라이언트에서 다시 이 RGB 값이 저장된 List를 읽어서 이미지로 복원한 후 클라이언트 컴퓨터의 화면에 표시하는 방식을 시도하였다. 그러나 이 방법은 대용량 데이터 전송에 최적화되지 않은 Pyro4로 전송할 경우 1024x1024 픽셀 정도의 데이터 전송에도 사용자 경험을 악화시킬 정도의 상당한 시간이 소요되는 문제가 발생한다. 또한 이렇게 생성된 이미지는 PNG 또는 JPEG 형식의 정적 이미지로서 그림 9에서 표시한 대로 복잡한 전력망의 경우 edge weight (무효 전력, 송전선 전력 용량 등)의 값과 node의 정보가 중복 표시되어 오히려 파악하기가 힘들다.

Pyro4는 이미지 등의 대용량 데이터 전송을 위해 별도의 함수를 제공[12]하고 있기는 하지만 정적 이미지로 전력망 정보를 표현할 경우의 한계는 여전히 해결이 힘들어서 그림 11에서 제시한 대로 node.js [13]와 웹 서버로 구성되는 새로운 레이어를 서버에 적용함으로써 보다 나은 방안을 구현하였다.

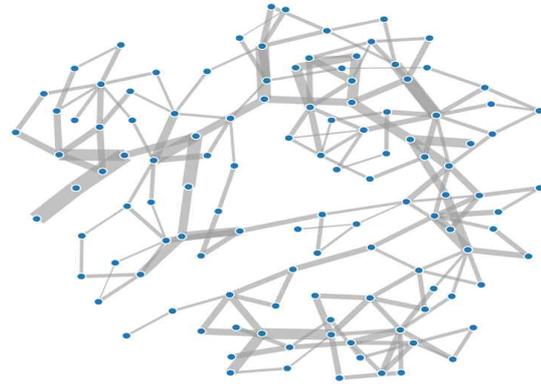


그림 10. d3.js 기반으로 구현된 동적 네트워크 그래프

Fig. 10. Dynamic Network Graph Implemented using d3.js

즉 서버와 클라이언트 사이에서 HTTP 프로토콜을 통해 웹 자원을 식별(URI; Uniform Resource Identifier)하고 최적의 전송 방법을 적용함으로써 클라이언트는 웹 브라우저로 서버가 제공하는 정적 이미지를 일반 웹 서비스처럼 빠른 시간 내에 전송 받을 수 있게 되었다. 더 나아가 JavaScript 기반 인터랙티브 정보 시각화 라이브러리인 d3.js [14]를 이용해 그림 10과 같은 동적 네트워크 그래프[15]로 웹 브라우저 상에서 전력망 정보를 인터랙티브하게 제공하였다. 논문 상의 그림 10에서는 동적 네트워크 그래프의 특성이 잘 나타나지 않으나 실제 웹 브라우저 상에서 전력망 알고리즘 개발자들은 파란색으로 표시된 각 노드들을 마우스로 드래그 해서 원하는 위치로 이동시킬 수 있다. 또한 기본적으로 edge weight의 상대적인 크기에 비례해서 edge의 너비가 표시되며 edge와 node에 마우스 커서를 위치시키면 edge weight와 node 정보를 수치 및 텍스트 정보로 팝업 도움말 형식으로 표시되도록 구현하여 전력망 알고리즘 개발자들의 사용성과 사용자 경험을 크게 개선할 수 있었다.

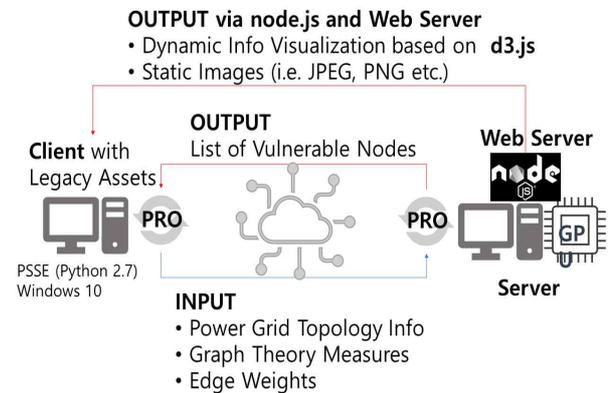


그림 11. node.js와 웹서버, 정보 시각화 라이브러리 d3.js를 이용한 웹 브라우저 내 동적 시각화 확장 구조

Fig. 11. Extended structure for dynamic information visualization on web browser based on node.js, web server, d3.js

표 1. 기존 전력 정보 관리 시스템과의 비교

Table 1. Comparison with Traditional Energy Management System (EMS)

	Traditional EMS	Proposed Architecture
Scalability	Hard to scale due to heterogeneous modules (e.g. Matlab, C/C++, Python, Fortran, etc.)	Scalable based on interfaces implemented with Python Remote Objects
Race condition	Extreme due to centralized database	Low thanks to stateless design
Usability	Low or limited due to static result images	High and extensible thanks to interactive web interfaces based on d3.js
Maintenance	Difficult due to different communication protocols and various development environments	Easy thanks to modular design and common interfaces implemented in Python

### III. 결 론

본 논문에서는 차세대 전력망 운영 시스템 개념 설계 및 개발을 목표로, 전력전공자들과의 협력 연구를 통해 구현된 차세대 지능형 분산형 전력망 분석 및 예측 기술의 구현 사례를 중심으로 개발 과정에서 상이한 전공 배경과 개발 능력을 가지고 있는 대학 연구팀들 사이의 격차를 실질적으로 극복하고 유연한 시스템을 설계, 구현한 사례를 소개한다. 특히 전력망 시스템은 안정적인 운영과 보안을 위해 오랜 기간 동안 검증된 레거시 시스템을 이용하는 경향이 있는데 차세대 전력망 개발을 위해서는 이러한 레거시 시스템과의 통합이 필수적이다.

본 논문에서 소개하는 구현 방법론의 핵심적인 요소는 개발 진입 장벽이 낮은 Python 프로그래밍 언어 및 Python Remote Object로 레거시 시스템과의 인터페이스를 구성하고 데이터베이스 및 중앙집중화에 과도하게 중점을 둔 기존 전력망 관리 및 분석 시스템의 구조를 분산화하며 동시에 RESTful API 제작의 한 축인 스테이트리스(stateless) 원칙을 도입하여 유연성과 확장성을 개선한다는 것이다.

표 1에서는 기존 전력 정보 관리 시스템과 제안하는 아키텍처를 비교한다. 기존 전력 정보 관리 시스템에 대해 제안하는 아키텍처의 장점은 다음과 같이 정리할 수 있다.

**1. 확장성 :** 기존 전력 정보 관리 시스템의 구성 모듈들이 서로 다른 개발 언어 및 환경으로 구현되어 서로 간의 통신과 연동 난이도가 높다. 제안하는 방식에서는 Python Remote Object를 이용해서 서로 다른 구성 모듈들의 공통 인터페이스를 구현하고 통신을 수행하도록 함으로써 프로그래밍 개발 능력이 상대적으로 부족한 전력 전공자들도 다른 모듈과의 통신 및 연동을 위한 기능을 쉽게 확장할 수 있도록 설계하였다.

**2. 동기화 문제:** 중앙집중화된 데이터베이스로 인해 동기화 문제를 해결하기가 어려운 기존 방식에 비해 제안하는 방식은 스테이트리스 설계를 적용함으로써 동기화 문제를 해결하였다.

**3. 사용성:** 기존 전력 정보 시스템은 웹 환경에 대한 고려가 부족하기 때문에 일반적으로 데이터의 시각화를 정적(static) 이미지로 한정하는 문제가 있다. 제안하는 방식은 웹 인터페이스를 제공하는 웹 서버를 구성 모듈들 중 하나로 활용(그림 11) 하기 때문에 d3.js와 같은 인터랙티브한 사용자 인터페이스를 구성함으로써 그림 10에서 제시한 것과 같은 동적 인터랙티브 시각화 정보를 제공할 수 있다.

**4. 유지 관리:** 기존 전력 정보 시스템에서는 모듈형 설계라고 하더라도 모듈 간의 독립성이 완전하게 보장되지 않거나 외부 모듈과의 연계가 고려되지 않은 경우가 많다. 이에 비해 제안하는 방식은 모듈 단위로는 원하는 방식(즉, 개발 언어, 운영 환경)을 자유롭게 선택할 수 있고 Python Remote Object로 공통 인터페이스를 구성함으로써 보다 독립적이고 연동이 강화된 분산 시스템을 구성할 수 있다.

본 논문에서 제안하는 방법론은 4차산업혁명에서 필수적인 타산업 분야와 IT 융합 개발 및 프로토타입 개발, 검증에서 직면하게 되는 문제를 해결하기 위한 실무적인 방안으로 적용 가능하리라 판단한다.

### 참고문헌

- [1] K. Ahn, "Study on the developmental directions of Mobile contents for Mobile e-learning prosumers in Web 2.0 decade", Journal of Digital Contents Society, Vol. 9, No. 1, pp. 41-51, 2008.
- [2] W. Han, "Alphagrid, a Smartgrid Considering People", Philosophical Studies, Vol. 0, No. 123, pp. 187-210, 2018.
- [3] J. S. Jo, "Patent Technology for Convergence of AI and Medical Technology Between Korean and Japan in the Fourth Industrial Revolution", Journal of Digital Contents Society, Vol.20 No.9, pp. 1,883 - 1,892, September 2019.
- [4] H.M. Seo and S.A. Kim, "A Study on Consumer Attitudes toward ICT Convergence Technology of Professional Sports Team: Focused on the Extended Technology Acceptance Model(ETAM)", Journal of Digital Contents Society, Vol.21 No.3, pp. 521 - 528, March 2020.
- [5] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine, 2000.
- [6] I. de Jong. Pyro4 Official Site. Available: <https://pypi.org/project/Pyro4/>.
- [7] T. Gonen, Modern Power System Analysis, 2nd Edition, CRC Press, 2013

- [8] Siemens, Power System Simulator for Engineering, Available:  
<https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/pss-software/pss-e.html>
- [9] NetworkX official site, Available: <https://networkx.org/>
- [10] M. Rubinov and O. Sporns, “Complex network measures of brain connectivity: Uses and interpretations”, NeuroImage, Vol. 52, No. 3, pp. 1059-1069, 2010.
- [11] D. H. Kim, D. A. Eisenberg, Y. H. Chun, J. Park, “Network topology and resilience analysis of South Korean power grid”, Physica A: Statistical Mechanics and its Applications, Vol. 465, pp. 13-24, 2017.
- [12] Pyro4, Binary data transfer / file transfer. Available: <https://pyro4.readthedocs.io/en/stable/tipstricks.html#binary-data-transfer-file-transfer/>
- [13] Node.js official site. Available: <https://nodejs.org/>
- [14] d3.js official site. Available: <https://d3js.org/>
- [15] Dynamic network graph. Available: <https://bl.ocks.org/matt-bernhardt/32ad5d557471f3a00703>



**박태정(Taejung Park)**

1997년 : 서울대 전기공학부 (공학사)  
1999년 : 서울대 전기공학부 대학원 (공학 석사, 반도체 물리 전공)  
2006년 : 서울대 전기컴퓨터공학부 대학원 (공학박사, 컴퓨터 그래픽스 전공)

2006년~2013년 : 고려대학교 연구교수  
2013년~2017년 : 덕성여자대학교 정보미디어대학 디지털미디어학과 조교수  
2018년~현재 : 덕성여자대학교 공과대학 사이버보안/IT미디어공학과 부교수  
※ 관심분야 : 컴퓨터그래픽스, 병렬처리, 인공지능, 수치해석, 3차원 모델링