Modbus/TCP 프로토콜 기반 클러스터링 알고리즘 성능 평가

이민성, 심규석, 이민섭, 박준상*, 김명섭
고려대학교, *LG Electronics
{min0764, kusuk007, chenlima2, tmskim}@korea.ac.kr, *junsang.park@lge.com

Performance Evaluation of Modbus/TCP Protocol-based Clustering Algorithm

Min-Seong Lee, Kyu-Seok Shim, Min-Seob Lee, Jun-Sang Park*, Myung-Sup Kim
Korea Univ., *LG Electronics

요 약

산업 제어 기술이 발전하고 있고 자동화 기술이 발전함에 따라 네트워크 통신 환경이 중요하다. 네트워크 통신을 위한 통신 프로토콜이 필수적이며 그에 대한 연구가 필요하다. 특히 산업자동화의 증가에 따라 산업 프로토콜의 가치가 증가하고 이에 따른 산업 프로토콜 리버스 엔지니어링이 필요하다. 트래픽에 관련된 여러 가지 연구들이 진행이 되고 있고 다양한 방법으로 트래픽을 분석하고 있다. 본 논문은 프로토콜에 대한 두 가지 클러스터링 알고리즘의 성능을 분석하고 비교한다. 클러스터링 알고리즘의 성능을 분석 후 향후 프로토콜 리버스 엔지니어링 연구에 사용될 적합한 클러스터링 알고리즘을 제시한다. 본 실험에서는 산업 제어 기술 통신을 위한 프로토콜 리버스 엔지니어링에 K-means Clustering이 적합하지만 향후 더 다양한 Clustering 방법을 적용해 볼 필요가 있다.

Ⅰ. 서론

산업 제어 기술이 발전하고 있고 자동화 기술이 발전함에 따라 네트워크 통신 환경이 중요하다. 네트워크 통신을 위한 통신 프로토콜이 필수적이며 그에 대한 연구가 필요하다. 기존에 사용되던 프로토콜은 비트 전송의 신뢰도가 떨어졌고, 그것을 향상시킨 산업용 이더넷 통신이 주목을 받게 되었다. Modbus/TCP 프로토콜은 Modbus에서 한 단계 발전한 버전으로 TCP/IP를 활용하여 Modbus 메시지 전송을 구현한다[1]. 본 논문은 다양한 방법 중 Clustering 알고리즘을 통해 Modbus/TCP 프로토콜을 분석하였다.

Clustering 알고리즘은 주어진 데이터의 특성을 고려하여 군집을 형성하고 그 군집을 나타낼 수 있는 점을 찾는 방법이다. 이러한 Clustering 알고리즘을 Modbus/TCP 프로토콜에 적용하여 군집을 어떻게 형성하고 그 군집형성이 프로토콜의 특성에 알맞게 분류가 되었는지 확인하며 두 가지 Clustering 알고리즘의 성능을 분석하여 다양한 프로토콜들에 대한 적합한 Clustering 알고리즘을 제안하고자 한다.

Ⅱ. 본론

본 장에서는 Modbus/TCP 프로토콜의 구조에 대하여 정의하고 K-means Clustering 알고리즘과 Mean-Shift Clustering 알고리즘과 분석할 실험 데이터에 대해 언급한다.

2.1 Modbus/TCP 프로토콜 구조

본 절에서는 Modbus/TCP 프로토콜의 구조에 대하여 언급한다. Modbus/TCP 프로토콜은 MBAP(Modbus Application Protocol) 헤더와 함수 코드(Function Code), 그리고 함수 코드에 따른 Data로 이루어진다. MBAP 헤더에는 Transaction ID, Protocol ID, Length, Unit ID로 이루어져 있다. Transaction ID는 Query 및 Response 한 쌍의 작업으로 구분하기 위해 사용되는 번호이며 마스터에 의해 설정된다. 최초 0x0000값부터 통신 시작 시 1씩 증가시킨다. Protocol ID는 Protocol의 ID를 나타내며 0x0000으로 고정 값이다. Length는 Length 필드 위치에서 프레임 마지막까지의 길이를 나타내며 Unit ID에서부터 Data 끝까지의 Byte 수이다.Unit ID는 TCP/IP가 아닌 다른 통신선로의 연결되어있는 서버를 구분하며 TCP 포트는 0x01로 고정이다. 함수 코드는 Modbus 프로토콜에서 제공하는 명령어 집합 코드로서 Memory에 값을 읽어오거나 쓸 수 있는 서비스이다. 본 연구실에서 수집한 Modbus/TCP 트래픽의 함수 코드는 90으로 고정이다. Data는 함수 코드에 따라 구조가 조금씩 달라진다. 본 논문에서 다룰 Clustering 알고리즘의 데이터는 Modbus/TCP의 Data 부분이다.

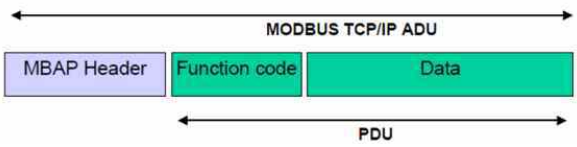

그림 1. Modbus/TCP 프로토콜 구조

표 1. MBAP 헤더

항목	길이
Transaction ID	2 bytes
Protocol ID	2 bytes
Length	2 bytes
Unit ID	1 bytes

2.2 Clustering Algorithm

※ 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2018R1D1A1B07045742)과 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-00539-001,블록체인의 트랜잭션 모니터링 및 분석 기술개발)

본 절에서는 성능평가에 사용되는 두 가지 Clustering 알고리즘에 대해 언급한다. 선정된 두 가지 Clustering 알고리즘은 K-means와 Mean-Shift 알고리즘이다.

K-means 알고리즘은 주어진 데이터들 다시의 거리 혹은 유사성을 이용하여 K개의 클러스터로 군집시켜주는 알고리즘으로 K는 구분하고자 하는 군집의 개수를 의미한다[2]. 이는 사용자가 임의로 원하는 군집의 개수를 정할 수 있다는 것을 의미한다. K-means 알고리즘은 사전 정보가 필요 없이 거리만 가지고 군집화가 가능하지만 군집의 개수를 사용자가 정해야한다는 단점이 있다. 실험에서는 elbow기법을 통해 군집의 개수를 정의한다.

Mean-Shift 알고리즘은 밀도기반 Clustering 알고리즘으로서 데이터의 무게중심을 찾아가는 방식이다[3]. 자신의 주변에서 원을 그리며 가장 데이터가 밀집된 방향으로 이동하며 수렴할 때 그 데이터가 군집의 중심이 된다. Mean-Shift 알고리즘은 평균이동이 자동으로 각 군집의 중심 데이터를 찾기 때문에 군집의 개수를 선택할 필요가 없다. 하지만 원의 크기(bandwidth)에 따라 군집의 개수가 달라진다. 실험에서는 Bandwidth의 quantile값을 권장값인 0.3으로 설정하여 실험한다.

2.3 실험 Data

본 절에서는 실험에 사용된 Modbus/TCP에서 Data 부분이다. Data는 함수 코드에 따라 달라지는데 기본적으로 Start Address, Length, Byte Count, Data의 형태를 가진다.

표 2. DATA

항목(길이)	설명
Start Address[2 Bytes]	접근하려는 메모리의 시작번지
Length[2 Bytes]	시작번지부터 값을 읽거나 쓸 길이
Byte Count[1 Bytes]	Request, Response에 따른 메모리 Data byte 수
DATA[N Bytes]	Request, Response에 따른 메모리 Data의 값

Ⅲ. 실험 및 결과

본 장에서는 선정된 클러스터 알고리즘을 사용해 실험을 진행한다. 실험과정은 실험에 사용할 Modbus/TCP의 Data를 추출한 후에 Size별로 구분 후 각 Size별로 두 개의 Clustering 알고리즘을 진행한다.

본 연구실에서 수집한 약 22개의 트래픽 셋의 Modbus/TCP의 Data를 추출하여 길이별로 구분을 한다. 총 7079개의 Data들 중 Size별 Clustering을 할 필요가 없는 개수가 적은 표본들을 제외하고 실험을 진행한다.

표 3. 실험 데이터

Data Length	개수	Data Length	개수
2	1327	22	35
3	599	28	96
4	78	49	42
6	684	68	705
8	69	105	226
9	210	161	134
11	1027	1019	99
12	365	1020	615
14	162		

본 실험에서 K-means와 Mean-Shift 방식의 Clustering의 군집화 개수는 크게 차이가 나지 않지만 K-means에서 군집도를 보다 더 정교하게 나타낸다는 것을 나타낸다. 실험결과의 한 예시중 Length가 1020인 Data Clustering을 실험한 결과, K-means의 경우 타입을 구분시 다음 결과와 같이 T로 가득찬 메시지와 00부터 FF까지 순차적으로 증가하는 메시지를 구분한다. 그러나, Mean-Shift의 경우 두 개의 타입을 구분해내지 못하기 때문에 현재 K-means가 산업 제어 기술 통신을 위한 프로토콜 리버스 엔지니어링 기술에 적합하다. 하지만, 향후 UPGMA와 Needleman-Wunsch등 더 다양한 Clustering 방법을 적용해 볼 필요가 있다.

표 4. 실험 결과

Length	K-means cluster	Mean-Shift cluster
2	4	4
3	8	8
4	5	4
6	4	6
8	3	2
9	5	4
11	4	8
12	4	4
14	7	2
22	5	4
28	4	2
49	6	4
68	6	6
105	7	4
161	4	4
1019	3	3
1020	4	1

표 5. K-means Type(Length : 1020)

Method	Type	
K-means	Type 0	"T,...T"
	Type 1	"00,01,02,...,FF"

Ⅳ. 결론 및 향후 연구

본 논문은 산업 제어 기술 통신을 위한 통신 프로토콜에 대한 두 가지의 Clustering 알고리즘을 실험하였다. 실험한 결과 향후 연구에 사용될 프로토콜 리버스 엔지니어링에는 K-means Clustering이 적합하다. 하지만 더 다양한 Clustering 방법을 적용해 볼 필요가 있으다. 또한, 산업 제어 기술 통신 프로토콜뿐만 아니라 사용되고 있는 다양한 프로토콜에 대한 Clustering 연구도 필요하다.

향후 연구로는 Netzob에서 사용중인 두가지 Clustering 방식인 UPGMA와 Needleman-Wunsch 알고리즘을 통한 실험을 진행 할 예정이며 상용 프로토콜에서 어떤 Clustering 알고리즘이 적합한 결과를 도출할 것인지 실험할 계획이다.

참 고 문 헌

[1] Denton, G., Karpisek, F., Breitinger, F., & Baggili, I."Leveraging the SRTP protocol for over-the-network memory acquisition of a GE Fanuc Series 90-30". Digital Investigation, 22, 2017, S26-S38.

[2] Jain, A.K, "Data clustering: 50 years beyomd K-means.", Pattern recognition letters, 2010, pp.651-666

[3] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on Pattern Analysis & Machine Intelligence 5, 2002, pp.603-619.

[4] G. Bossert, "Exploiting semantic for the automatic reverse engineering of communication protocols," Ph.D. dissertation, Univ. Gif-sur-Yvette, Rennes, France, Dec. 2014.