CIGRE-284 **2020 CIGRE Canada Conference**
Toronto, Ontario, October 19-22, 2020

Auto-DR: High performance communication architecture for automatic acquisition of disturbance record at control centre

R. K. ARORA, P.K. SRIVASTAVA, SUNIL KUMAR, MANOJ KUMAR, NITIN SINGH, JOSEPH GEORGE JOSE
Power Grid Corporation of India Limited
India

SUMMARY

In power system fault analysis, Disturbance Record (DR) is an important tool to reliably investigate disturbance after the fault has occurred. Disturbance recording is done by disturbance recorders or by IEDs (Intelligent Electronic Device). Traditionally, these disturbance records were collected locally in the substations and were analysed manually to find the root cause of the fault. In recent years, various solutions have come up to remotely acquire DR from control centre for centralized analysis of DR which enables operators to expedite corrective action. However, these remote DR acquisition solutions require proprietary hardware and a number of them do not scale well when load increases which affects its performance.

This paper proposes a new architecture, called Auto-DR which is implemented in POWERGRID, for acquiring DR in a reliable, secure and time efficient manner along with discussing its various benefits. It introduces a pull mechanism of DR acquisition in which control centre software actively retrieves DR from substation as opposed to substation hardware/software pushing DR to control centre systems.

KEYWORDS

Disturbance record, DR remote acquisition, High performance, Scalability, Security, Reliability, Communication architecture, Worker node, AFAS (Automated Fault Analysis System)

rkarora@powergridindia.com

1 Introduction

POWERGRID is public sector transmission utility which accounts for transmitting about 50% of total power generated in India. This architecture is implemented as part of POWERGRID's flagship project National Transmission Asset Management Centre (NTAMC) which allows for remote operation of substations from control centres. Automated Fault Analysis System (AFAS) is part of NTAMC project, which analyse DR from substations to identify nature of fault and its location and generates notifications to operators and other stakeholders. Auto-DR architecture focuses on delivering DR to AFAS system by collecting from substation.

While various solutions exist to transmit DR from substation to control centre, most of them rely on substation hardware/software to send DR to control centre. It is sensible to adopt this strategy when deployed in a setup with fewer numbers of substations. However, as the number of substations increases, volume of DR to be collected also increases and this methodology will struggle to keep up with the demand of collecting DR in stipulated time.

In the Auto-DR architecture proposed and implemented in this paper, worker nodes at control centre pulls DR files from substation upon intimation of new DR by substation. As worker nodes are less expensive in terms of computer resource usage, more worker nodes can be deployed to process large numbers of DR collection requests without affecting system performance. Our implementation uses open-source software which runs on generic operating systems without any need for proprietary hardware.

The responsibility to extract DR from IED/relay at substation is delegated to third-party/OEM software solutions which communicate with relays to extract DR automatically either on polling or event triggered basis. The collected DR is stored in a predefined location. Auto-DR architecture implementation integrates well with these standard solutions to collect the DR from the stored location and focuses its effort on transmitting the DR to control centre in a reliable, time efficient and secure manner.

2 Components of Auto-DR architecture

Auto-DR architecture implements a pull mechanism to collect DR from substation in which control centre software components actively collects DR from substation whenever new DRs are available. This methodology provides better time efficiency in collecting DR as time required for each substation is less because of multiple worker nodes. This is in contrast with the push methodology where in substations push the DR to control centre in which substation may have to wait its turn for the control centre software to be ready for that substation.

Auto-DR architecture consists of set of components at control centre and substations. The components at substation are deliberately designed to be simple compared to control centre counterparts. This architecture is following a modular approach for achieving ease of configuration and maintainability. The architecture is depicted in figure 1.

2.1 Substation level components

There are two components at substation which are Auto-DR poller and Auto-DR manager. The poller component interfaces with OEM application to collect DR. Manager and poller is together referred as Auto-DR client. The following sections explain each component in detail.

2.1.1 Poller

This software component runs on the substation gateway device which is connected to both control centre and DR/engineering workstation, where DR is collected, through computer networking. As the name suggests, the poller polls for new DR from the particular folder where the third-party/OEM vendor software stores the DR collected from IEDs. The method of polling depends on the protocols supported by the DR/engineering workstation and it can be through protocol such as Server Message Block (windows sharing), File Transfer Protocol (FTP), etc. The method of polling can be configured at runtime of the component which provides flexibility in choosing different protocol.

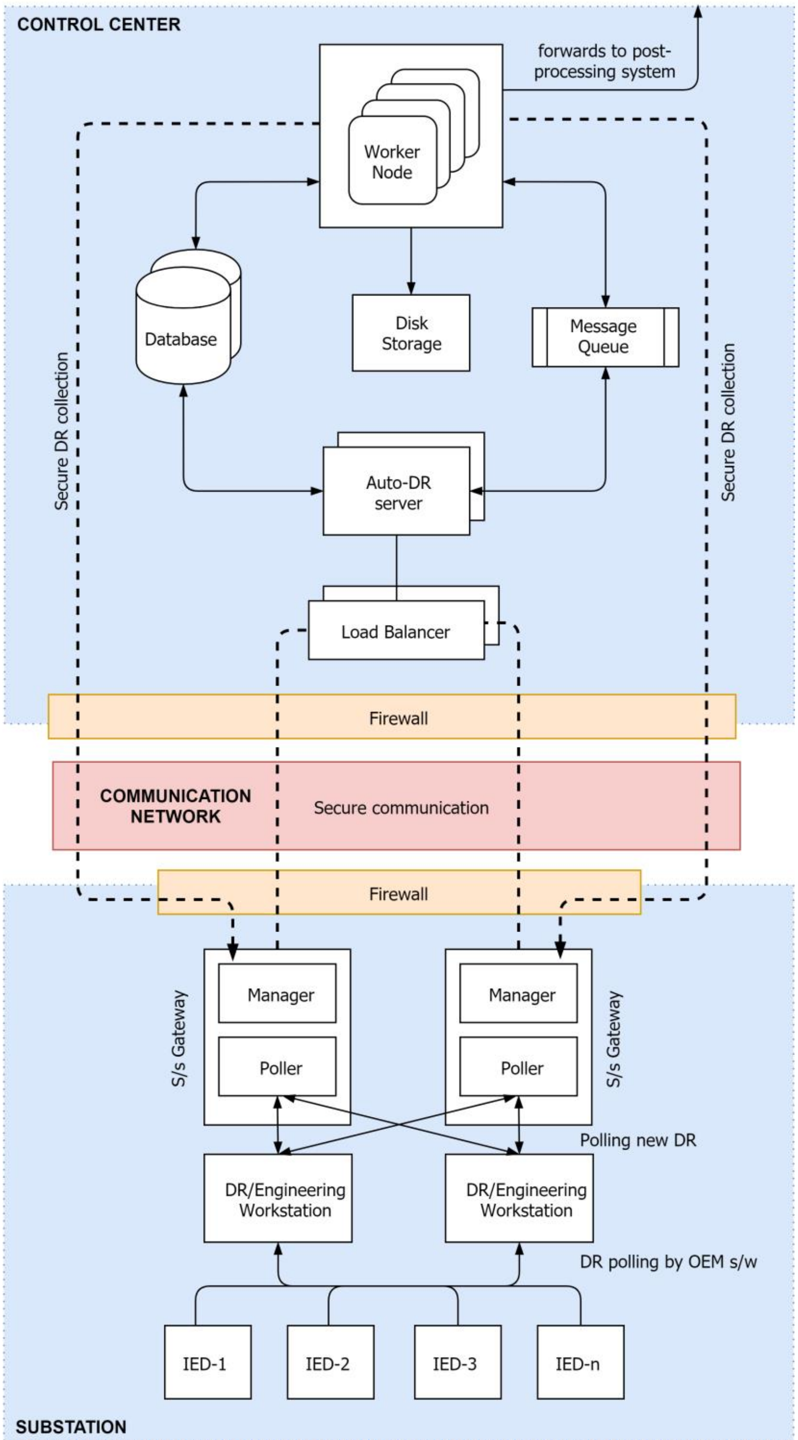

Figure 1. Overview of Auto-DR architecture

Poller expects DR files to be in COMTRADE format and major OEMs support COMTRADE format for DR in their IEDs. A DR in COMTRADE format contains a set of files for each fault in which CFG and DAT files must be present at the minimum, in addition to other files such as HDR etc. However, some OEMs create separate files for each DR whereas others bundle related files of a DR in zip format and some may have other bundling formats. By following a design pattern known as observer pattern, poller component can handle these varying requirements of DR file collection by various OEM efficiently by deploying different observer for each case. This manner of handling varying requirements by separating concerns to different observer helps in maintenance and troubleshooting as well.

Poller expects the DR files collected by OEM software in separate folders for each relay. By using a pre-configured mapping of relay names to folder names, it can rename the COMTRADE files accurately as per naming convention, defined in the IEEE standard C37.232, and forward to Auto-DR manager which will send to Auto-DR server. Poller keeps track of collected files in local storage so that any failure, to the poller component or to the machine running it, will not affect the functionality of poller.

This component is referred as Auto-DR poller or simply "poller" for the remainder of paper.

2.1.2 Manager component

The manager component acts as a bridge between Auto-DR server and poller. Whenever poller gets a new DR it is forwarded to manager. In addition to sending DR files from poller to Auto-DR server it performs various other functions such as the following.

1. high availability by communicating with peer managers
2. Authentication with Auto-DR server and storing session token
3. Monitoring of poller.
4. Maintaining heartbeat data to inform Auto-DR server of its health status
5. Retrying failed communications with Auto-DR server
6. Storing operation state to disk to achieve fault tolerance in case of any system or component failure
7. Validating worker nodes using session token
8. Enabling maintenance mode of manager and poller components to not collect DR during testing or maintenance.

This component is referred to as Auto-DR manager or simply "manager" for the remainder of paper.

2.2 Control centre level components

In this architecture, the intricacies in achieving high performance is pushed to the control centre level so that components at substation can remain simple. However, the complexities involved are segregated according to their functionalities which is called the principle of separation of concerns. To achieve this, various key software components are used which are explained below.

2.2.1 Load Balancer

Load balancer is a device or software which distributes network or application traffic across a number of servers. In this architecture, load balancer is used to distribute traffic to multiple instances of Auto-DR server either from substations or other internal components. Through the use of load balancer, impact of failure of some of Auto-DR server instances is reduced thereby increasing availability of server. Another important function of load balancer is to act as a reverse proxy; only load balancer IP will be exposed and the internal IP of Auto-DR server will be abstracted away from outside entities, thus improving security.

2.2.2 Auto-DR server

Auto-DR server handles the core application logic of this architecture. It is a web application which co-ordinates DR collection from substation. Various functions performed by Auto-DR server is as follows:

1. Initiating the DR collection workflow, explained in section 3.1, on receipt of request from substation Auto-DR manager.

2. Configuration management of Auto-DR manager at substation which includes providing list of allowed hosts, enabling/disabling maintenance mode.
3. Monitoring the health of substation components using periodic heartbeat signals.
4. Assigning works to worker nodes by pushing DR collection requests to message queue.
5. Handling authentication of Auto-DR manager by issuing session token
6. Storing transaction logs in database
7. Archiving DR in persistent storage disk for future use.

2.2.3 Message queue

In this architecture, DR collection and post-processing tasks are done in an asynchronous fashion. In asynchronous processing of tasks, constituent components do not wait for each other to finish, while performing its assigned task. In doing so, it is imperative that these components are communicating well with each other to avoid any processing delays. To ensure, instant communication among the components, a message queue application is used in this architecture.

As message queue constitutes a major part of the architecture handling communication among control centre components, it is essential that message queue software supports the following features.

1. Fault tolerance: it is capable recovering from system and software failures
2. High performing: it is capable of relaying messages with least delay.
3. High availability: multiple instances are capable of operating in a clustered manner and supports high available configurations through load balancers.
4. Security: it supports security features such as authentication, encryption.

With these features in place, Auto-DR server and worker nodes can reliably communicate with each other. Whenever high volumes of DR files are to be collected, the number of outstanding messages in the queue will be more. In this situation more worker nodes can be spawned automatically to process the backlog of requests. This way, the number of worker nodes can be scaled up or down based on the queue size, hence achieving high performing and efficient collection of DR or forwarding DR to post-processing system such as AFAS.

2.2.4 Database

Database technology has matured well, over the years, and various database software solutions are available which supports redundant configurations across multiple machines along with synchronization among the instances. In this architecture, we use a relational database application for persistent and reliable storage for transaction logs and to store other system information.

2.2.5 Worker nodes

The term worker node refers to a small piece of software which does only one pre-configured job well. Worker nodes forms an integral part of the asynchronous methodology used in this architecture.

There are two types of worker nodes used in this architecture; first type to pull DR from substation Auto-DR manager and another type to forward the collected DR to the post-processing system. While collecting DR from substations, it collects relevant session token from database to inform substation manager that it is authorized for this activity. Worker nodes can be created and destroyed based on the performance demand from time to time, i.e. when more requests are pending in the message queue more workers can be created to process these and can be destroyed otherwise.

3 Detailed workflow

This section explains detailed workflow of collecting DR from IEDs, sending to control centre and forwarding to post-processing systems for analysis of DR. This workflow is shown in figure 2.

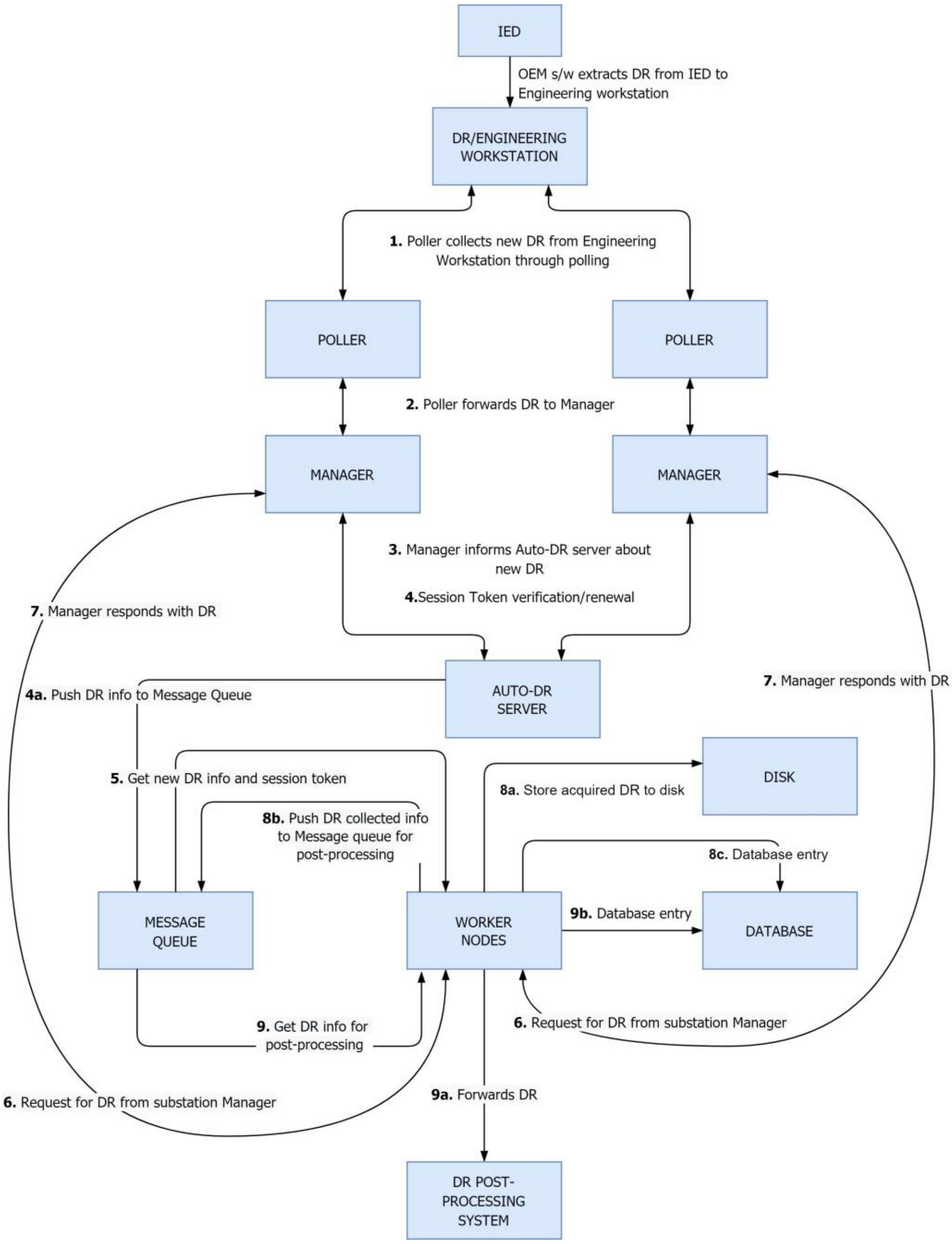

Figure 2. Detailed workflow of Auto-DR architecture

3.1 DR retrieval mechanism

1. Auto-DR Poller collects DR as explained in section 2.1.1.
2. Auto-DR Manager is subscribed to the information of new DR collection by poller and as soon as DR arrives in poller, manager collects the DR files from poller and stores in the local storage.
3. Auto-DR manager informs Auto-DR server about the availability of new DRs along with details of DR. Any redundant information, because multiple managers reporting the same information, will be preserved until the DR is collected by worker nodes. Auto-DR manager has to present a valid session token to Auto-DR server while informing about DR.

4. Auto-DR server verifies token of the request.
    a. If token is valid, server pushes the new DR information to message queue.
    b. If token is invalid, server informs Auto-DR manager to request for a new session token. Tokens have a predefined validity period, so new session token can be obtained by the Auto-DR manager when it is idle before the token expires.
5. As soon as any new DR information arrives, one of the worker nodes subscribed to this information, pops the message from queue. The message contains session token of Auto-DR manager which provided the DR information
6. Using this information and session token, worker node initiates a request to acquire DR from Auto-DR manager which informed about the new DR.
7. Auto-DR manager provides the worker node with requested DR files. If manager is unable to provide the DR files requested or if any failure occurs in the process, the transaction is recorded as failed in the database with details of the DR such as name, timestamp, substation detail, and relay details for any further analysis. However, as all Auto-DR managers send the DR information to server, some other worker node will collect the DR from substation there by repeating the process from step 5.
8. After successfully retrieving DR from substation, the worker node performs the following steps
    a. Stores collected DR to the designated disk storage
    b. Add an entry in the message queue with the details DR and its location in local storage.
    c. A database entry is made with details of the successful transaction.
9. Some of the worker nodes are subscribed to the message queue to get the message of successful collection of DR files. Once the message queue receives a successful DR collection message, it dispatches the details of collected DR to one of these workers which then performs the following steps.
    a. Retrieves the DR stored in disk and forwards to any post-processing systems which in our case is Remote Accessibility System (RAS) and Automated Fault Analysis System (AFAS) which processes the DR to identify nature of the fault and generates notifications.
    b. On successfully forwarding to the post-processing systems, the worker node makes a database entry with details of the DR.

If worker confirms that assigned job is completed, then it is removed from message queue. However, if any of the worker nodes fails to confirm task completion to the message queue within stipulated time due to worker node crash or any other reason, the said job will be re-assigned to another worker.. This cycle will continue until pre-defined number of times after which the job is declared failed and an entry is added to database.

3.2 Fault tolerance and high availability

Auto-DR architecture is distributed software solution across control centre and various substations. The distributed nature of architecture requires careful planning for ensuring high availability and fault tolerance. There are various measures considered in this architecture, both at control centre and substation level, to achieve these outcomes which are explained below.

3.2.1 Substation level

a. Manager and poller can work independently so that failure of one component does not affect the other one.

b. Multiple instances of Auto-DR client can be run simultaneously, either in same machine or multiple machines, which increases the availability of the application. Each Auto-DR manager component reports separately to Auto-DR server which increases availability of managers.

c. Manager and pollers keeps tracks of its state in local disk storage so that when system is recovered after crash or component failure, it can resume operation from the previous state. This also means that the manager will attempt to resend any failed DR file transmission to Auto-DR server for a defined number of times.

d. Any new DR files collected during a telecommunication link failure to control centre will be sent automatically after the link is re-established which improves fault tolerance.

3.2.2 Control centre level

a. Multiple instances of Auto-DR server component are run to improve availability and the load is distributed among these instances by using a load balancer.

b. The use of asynchronous processing through message queues along with worker nodes greatly improves the high availability and fault tolerance aspect at control centre. Worker nodes keep on consuming messages from message queues until there are no more messages. The message queue will remove a message from the queue only if it is acknowledged which occurs after successful processing by a worker node. The message will be re-queued if acknowledgement timeout expires or if a worker rejects message. This cycle of queuing and processing ensures continuous collection of DR and post-processing as long as message queue application is up. Hence by using a robust message queue application, this setup can withstand failure of multiple worker nodes and continue to function with at least one functioning node.

c. Multiple instances of database is spawned for its high availability and synchronization is configured among them to achieve fault tolerance of this component.

d. By running the application in multiple machines, hardware redundancy is easily achieved as all components support distributed mode of operating.

3.3 Security

The various security mechanisms considered in this architecture are as follows.

1. Secure channel for communication: In this architecture, all of the communications between control centre and substations are secured by using HTTPS protocol.

2. In this paper, session token mechanism is used for authenticating a client with server and a worker node with a client. Session token is issued by Auto-DR server to Auto-DR manager only if the substation IP address is allowed in the access list.

3. Access lists are used to restrict requests only from authorized clients and additional restrictions can be imposed at perimeter firewalls of control centre and substations.

4. The use of load balancer as a reverse proxy hides the actual identity of server where Auto-DR server is running.

4 Benefit

The following benefits are achieved through Auto-DR architecture to collect DR from substations.

1. Time efficiency: the ability of Auto-DR server to increase the number of worker nodes based on the load ensures timely DR collections from any substations with new DR, typically within 5 minutes.

2. High availability and Fault tolerance: Independent working of components, Mechanism to retry failed communication, Saving operational state to resume in case of failure, redundancy in hardware and software makes the architecture highly available and fault tolerant
3. Security: Authentication and encryption are implemented for communication between substations and control centre.
4. Scalability: The independent nature of substation and control centre components along with the use of message queues for asynchronous processing enables to scale up and scale down resources based on performance requirements
5. Configuration flexibility: The use of separation of concerns principle to modularize poller component helps it to support different protocols while polling for new DR.
6. Interoperability with other vendors: Poller at substation can interface with different OEM s/w to collect DR from IED.
7. No requirement for proprietary hardware: Auto-DR architecture utilizes the same substation reporting infrastructure used for remote monitoring/control of substations and any additional hardware is not required.

5 Conclusion

The Auto-DR architecture proposed in this paper provides an alternate and efficient way to remotely acquire DR in a reliable, secure and timely manner. The design principles used in this architecture such as separation of concerns and asynchronous methodology helps in achieving high performance, fault tolerance and flexibility with which it can integrate with existing DR collection infrastructure at substations. The use of existing hardware to implement the Auto-DR solution without requiring proprietary hardware makes this architecture cost-effective also.