VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ENHANCED ROBUST DECENTRALIZED PROBABILISTIC NETWORK MANAGEMENT

T. Peer Meera Labbai, Ankit Shukla, Aratrika Ganguly and Udbhav Agarwal
Department of Computer Science and Engineering, SRM University, Kattankulathur, Chennai, Tamil Nadu, India
E-Mail: u.agrawal30@gmail.com

ABSTRACT

The aim of this paper is to discuss an Enhanced Robust Decentralized Probabilistic Network Management using probabilistic and in network techniques. Methods: Management of distributed networks becomes difficult accounted by its ever-increasing size, complexity and pervasiveness; which makes it difficult to model accounting for its dependencies. In particular, we introduce an algorithm for peer-to-peer metric propagation, which combines the results of partial fusion from the locally made probabilistic models consistently, which help to mitigate overheads incurred in dynamic distributed systems and related redundant information gathering and processing.

Keywords: network, robust, decentralized, probabilistic.

INTRODUCTION

A distributed system is a collection of sites distributed over a network which communicate and coordinate their actions by passing messages[1]. The most essential characteristics of a distributed system are resource sharing, concurrency, fault tolerance and transparency which helps us to effectively manage the increasing complexity in network services in the recent years. As the large scale distributed system are complex it is impossible to build the hardware and software components that are mutually dependent on each other, thus we move from centralized to peer to peer approach[2]. The peer-to-peer approach in distributed systems enables to share data and resources on a large scale by eliminating the use of separately managed servers and their corresponding infrastructure[3]. We generate an algorithm for inter-agent belief propagation, to combine partial fusion results from local probabilistic models consistently, we show how to deal with the excess cost of superfluous information stored and processed, the management in a decentralized manner in dynamic and un-predictable environments[4], and the significant effort required for coordination of management functions in a decentralized way[5]. An unique clean-slate approach is formulated for the future of distributed systems by demonstrating how this framework can be applied to a network of information, an information centric distributed system with reduced resource management efforts. Fault tolerance in distributed systems is the ability of the system to continue to function properly even after partial failure, which means failure of any of the systems in the network[6]. To develop a distributed system this is hundred percent fault tolerant is practically challenging. Fault can occur because of node failure or malicious error. Fault tolerance provides the main features of distributed systems which are reliability, availability and security[7]. A fault is detected by constantly monitoring the performance of the system and comparing it with the expected outcome. If there is a deviation between the expected outcome and the actual outcome then a fault is reported. Fault diagnosis is done to understand the nature of the fault and the possible root cause. Report is generated based on the outcome of the fault diagnosis, which is referred to as fault evidence generation[8]. When a fault occurs, several alarm notifications are generated by the different components in a distributed system. Hence fault evidence is ambiguous as the same alarm may be generated for a different fault. It is inconsistent as a system may interpret a component to be faulty while the other system may interpret it to function properly. It may also be incomplete as a result of delay in sending the alarm notifications or the loss of the notifications. Therefore a fault management system should be designed which takes into account the ambiguous, inconsistent and incomplete information and provide a standard view of the network services[9]. A fault management system provides ways to interpret and represent uncertain data which is under the domain of the system and there is an evidence of the fault which has occurred[10]. Alarms can be set off by a fault which can be dependent on many factors such as the way the distributed systems are dependent on one another, the present configuration of the system and the services in use after the fault has occurred, etc. Because of these reasons, the knowledge of the system can be inconsistent and inaccurate[11]. As the alarms are over sensitive, they can be set off for even meager problems. Thus the system can be inconsistent as they cannot predict for what degree of fault has the alarm been set off and which alarm should be taken into consideration for the fault localization process. Hence this process is time consuming in nature. Fault management should deal with latent and complex dependencies. It is assumed that when a certain feature fails[7], the components that depend on this feature fail as well. In distributed system, such components are latent or hidden. Thus the strategy of event management is to build a complete system with an opportunity to manage each and every component of the system[12]. So a fault management system is required in order that one node should realize and display the fault occurred in the other connected node which is hidden. This is another benefit of peer-to-peer network.

VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

METHODOLOGY

A supernode - node network structure with In Network Management was considered, it allows different autonomous management techniques ranging from manual to full autonomous processes. In-network management also works along the line of abstraction, autonomicity and orchestration. In abstraction, it majorly deals with the reduction of external management interactions which gives a direct impact in reducing manual interaction and sustains the working of a large networked system[1]. For automaton, the management operations done manually are used for working with management parameters like manual routing configurations. For autonomicity, the system governs its own behavior in terms of network management, which helps in increasing the speed of the processes. Thus we use in-network management model in peer to peer connection because it helps to extend the system and its level of complexity to support the extension. Extension can be addition of a new functionality, new characteristics or modifications of existing characteristics. All these are need to be done by minimizing the impact on the existing system and also by making it a fast to increase the overall efficiency of the system[13]. In traditional system, all the nodes are individually connected to a server where the requests are getting processed, analyzed and then the required action is taken depending upon the analysis[14]. But in our model we propose a P2P architecture where the messages are passed between every node and requests are processed at every node and then finally sending the notification to the server. Thus we use in-network management and probabilistic model in a peer to peer architecture as it increases the efficiency of the system by a fair margin thus saving a lot of time.

MATHEMATICAL MODEL

n	Peers in community
s	Groups in community
M	Count of files in system
α	Zipf coefficient for file popularity
B_i	Files held by i^{th} peer
γ_{im}	Chance that i^{th} node shares file m
μ_0	Supernode service rate to search
μ_ι	Regular peer node service rate for provision
λi	Peer node i request rate for content
ρ	Degree for dispersion of peer positions
θm	Probability to ask for a file by a new request
$X_{i,m}$	File F_m is available which is stored in node i
$X_{i,m}$	File F_m is available which is shared on node i

Content provision distribution

θ_m follows a Zipf-like distribution

$$\theta_m = m^{-\alpha}.\theta_0$$

θ_0 is the normalization for zeta distribution
Provided α is positive.

If α = 0 it is uniform distribution for popularity of files.

$$\text{Also } X_{i,m}(\overline{X_{i,m}}) \begin{cases} 1 & \text{, if node}_i \text{ shares file } F_m \\ 0 & \text{, for any other case} \end{cases}$$

Where i represents the domain of nodes and m represents domain of files.
Let,

$P_{i,m} = E(X_{i,m})$
$\overline{P_{i,m}} = E(\overline{X_{i,m}})$

We know that

$X_{i,m} > \overline{X_{i,m}}$
$P_{i,m} > \overline{P_{i,m}}$

Assumptions

$P_{i,m} = \beta_{i,m} . \theta_m$

$\beta_{i,m}$ represents the intensity of content availability

$\beta_{i,m} . \theta_m$ belongs to [0,1]

$\beta_{i,m}.\theta_{1 \leq 1}$
$\overline{P_{i,m}} = P_{i,m} . \gamma_{i,m}$

Content request distribution
Assumptions follows Zipf-like distribution

$\lambda_{i,m} = \theta_m \cdot \lambda_i$

Search and provision process

Process is assumed to be exponential.
Service rate = μ_ι
Modeled as M/M/1 queue

Transmission delay distribution

By Global Network Positioning (GNP) approach

$T_{i,j} \in [0, \rho\pi]$

Provision policy

$j^* = \arg\min_j T_{i,j}$ s.t. $\overline{X_{j,m}} = 1$

Network availability

Content availability

$H(n,m) = 1 - \prod_{i=1}^{n} (1 - \overline{P_{i,m}})$

VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Overall expected content availability

$$H = E(H(n,m)) = \sum_{m=1}^{M} \theta_m$$

Search delay

$$\Lambda_0 = n \cdot \sum_{m=1}^{M} (1-P_m) \cdot \theta_m \cdot$$

Provision delay

Assuming j^{th} node shares m file with i^{th} node

$$= P(X_{i,m} = 0; \overline{X_{j,m}}=1; T_{i,j} < T_{i,k}, \forall k \neq i,j, s.t. \overline{X_{k,m}}=1)$$

$$= (1-P_m) . \overline{P_m} \sum_{k=0}^{n-2} 1/(k+1) . \binom{n-2}{k} \overline{P_m}^{k} (1-\overline{P_m})^{n-2-k}$$

$$= (1-(1-\overline{P_m})^{n-1}) . (1-P_m) \; /(n-1) \quad \text{(Binomial Theorem)}$$

$$\Lambda_j = \sum_{i \neq j} (\sum_m \lambda_{i,m} . Z(i,j,m))$$

$$= \sum_{m=1}^{M} \theta_m . (1-P_m) . (1-(1-\overline{P_m})^{n-1}) . \lambda$$

Finally,

$$D = (\mu - \Lambda_j)^{-1}$$

Transmission delay

$$T_{min}(k) = \int_0^{\tau} t\,k \, (1-F(t))^{k-1} f(t)\,dt = \tau/(k+1)$$

$$T_{min}(n,m) = E(T_{min}(k) \mid F_m \text{ is available})$$

$$= \frac{1}{\pi(n-1,m)} \sum_{k=1}^{n-1} T_{min}(k) . \binom{n-1}{k} \overline{P_m}^{k} (1-\overline{P_m})^{n-1-k} . \tau$$

Finally we have

$$T = E_m(T_{min}(n,m)) = \sum_{m=1}^{M} \theta_m . T_{min}(n,m)$$

Benefits analysis

1. There is an increase in content availability as the scale of the P2P networks increases.
2. Transmission delay decreases.
3. Content intensity increases which leads to an increase in the performance of the P2P network.
4. There is an increase in the level of sharing among the peers.
5. Position dispersion increases.

Disadvantages

1. Search delay increases.
2. Provision delay increases.

SIMULATION AND RESULT

A peer to peer in network based simulation was build using Java Simulator with aid of JavaBayes and BayesNetwork API in which each node was modeled with a parameter flag activated by a pseudo random number generator to avoid processing overheads. The likelihood of behavior of each node was monitored and it revealed that nodes behaved in almost similar fashion when the above-depicted probabilistic measurement was implemented. The equations in the above mathematical model were implemented to analyze the scaling effects in a network. The below graphs show how the content availability, transmission delay and the provisional delay varies based on the probability given to the node by the proposed mathematical model.

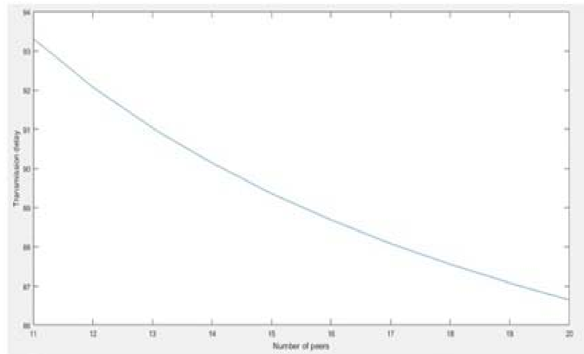

Figure-1. Variation of content availability with the Zipf coefficient.

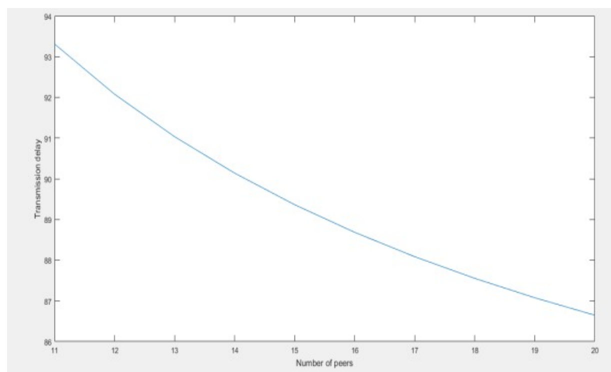

Figure-2. Variation of transmission delay with number of peers.

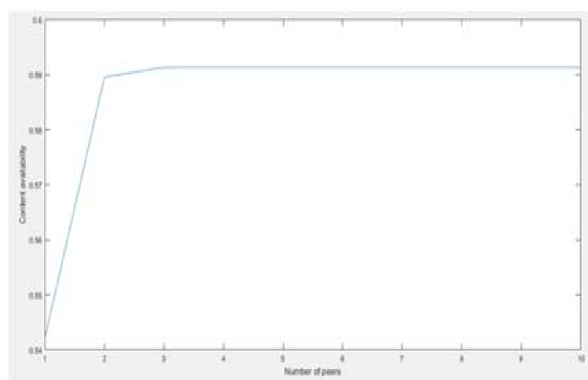

Figure-3. Variation of content availability with number of peers Fig 1.3 variation of content availability with number of peers.

VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

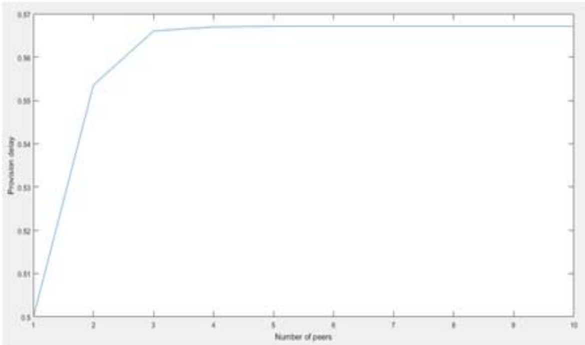

Figure-4. Provision delay with number of peers (λ=4).

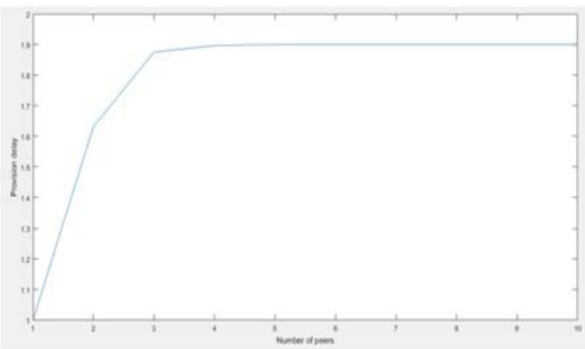

Figure-5. Provision delay with number of peers (λ=8).

CONCLUSIONS

As a conclusion, the simulation achieved the desired objectives. Thus if a network is established on decentralized peer to peer structure, the fault tolerance of system can be greatly maximized with inclusion of probabilistic In Network Management. This would ensure minimum processing overhead on system if are triggered with pseudo number generator to enable randomized measurement of network so that the dynamic nature of the network avoids any fault inclusion to the measurement of network metrics or effect its network transfer capabilities.

REFERENCES

[1] A Pras, B-J van Beijnum, R Sprenkels. Introduction to TMN.CIIT 1999.

[2] Mary Subaja Christo and S. Meenakshi. Reliable and Authenticated Rumor Riding Protocol for Unstructured Peer-to-Peer Network. Indian Journal of Science and Technology. 2016 June; 9(21):1-9.

[3] Jafari Navimipour, N. Sharifi Milani, F. Peer-to-Peer Netw. Appl. (2015) 8: 474

[4] Takeda A, Chakraborty D, Kitagata G, Hashimoto K, Shiratori N. A new scalable distributed authentication for P2P network and its performance evaluation. WSEAS Transactions on Computers. 2008 Oct; 7(10):1628- 37.

[5] M. Siegal and K, Sauer, "Efficient network management for a highly distributed system", Proc. of IFIP TC6/WG6.4 2nd International Conf. on Local Communication Systems, Palma, pp42 1-43 1, June 199 1.

[6] Cristian F. (1991) Understanding fault-tolerant distributed systems Com- munications of the Association for Computing Machinery 34(2):56-78.

[7] Sari, A. and Akkaya, M. Fault Tolerance Mechanisms in Distributed Systems. International Journal of Communications, Network and System Sciences. 2015; 8:471-482.

[8] Su, Lili; Vaidya, Nitin H. Asynchronous Distributed Hypothesis Testing in the Presence of Crash Failures; Feb 2016.

[9] Dupuy A, Schwartz J, Yemin iY,Barzilai G,Cahana A. Network Fault Management: A Users View in Integrated Network Management IV. Springer Publication, 1989; 101-107.

[10] Nirmala Devi, K.; Tamilarasi, A. Dynamic Scheduling in Grid Environent with the Improvement of Fault Tolerant Level. Indian Journal of Science and Technology, [S.l.], p. 507-515, apr. 2015. ISSN 0974 - 5645.

[11] Deng R H, Lazar A A, Wang W.A Probabilistic Approach to Fault Diagnosis in Linear Lightwave Networks. IEEE Journal on Selected Areas in Communication. 1993; 11(9), 1438-1448.

[12] A. T. Bouloutas, S. B. Calo, A. Finkel, Irene Katzela. Distributed fault identification in telecommunication networks. Journal of Network and Systems Management. 1995; 3(3), 295-312.

[13] E. P. Duarte *et al*., Improving the Dependability of Network Management Systems, Intl. J. Net. Mgmt., vol. 8, no. 4, July/Aug. 1998, pp. 24453.

[14] B. Gedik and L. Liu, Peer CQ: A Decentralized and Self Configuring Peer-to-Peer Information Monitoring System, Proc. Intl Conf. Distributed Computing Systems, 2003.