Network Transform – A Tool Based Approach

Shameemraj M. Nadaf, Hemant Kumar Rath, and Anantha Simha

***Abstract*—To minimize the cost of current operation and future deployment, consolidation or transformation of existing Information Technology (IT) elements is necessary. With IT Transformation of existing servers of one Data Center (DC) or a group of DCs, Network Transformation becomes crucial, such that appropriate bandwidth sizing and switch selection can be performed. So far this process is manual and vendor dependent. Therefore, we propose a Java based semi-automatic tool to provide a solution for Network Transformation and Optimization. We also discuss the performance of the proposed tool through a couple of practical examples and demonstrate significant improvement in cost of deployment and operation post transformation. Along with network transformation, this tool can also be used to analyze chattiness of applications/ports of the existing (*As-Is-State*) as well as transformed (*To-Be-State*).**

***Index Terms*—Data center consolidation, server consolidation, network transformation, network optimization, it consolidation, LAN, wan, optimization.**

I. Introduction

Major challenge faced by the CIOs these days is to run IT operations as a profit center rather than as a cost center. This requires CIOs to utilize their existing systems fully or do more with existing systems before going for newer ones. CIOs also have to find ways by which they can consolidate or transform multiple Data Centers (DCs) to fewer ones and may also consolidate the servers/storage devices present in a single DC. Consolidation helps to bring down the server sprawl, power consumption, maintenance etc., so that the cost of operation is reduced. Thus, IT transformation to create a more robust, scalable and cost-effective infrastructure from the existing one becomes essential. With consolidation of servers/storage, the network elements within a DC or across DCs must also undergo a consolidation and the Wide Area Network (WAN) must be optimized so that the user experience in terms of performance, up-time etc., remain same or improve as compared to that of the non-transformed network (*As-Is-State*). In this paper, we propose a semi-automatic network transformation tool which we have designed and developed using Java. This tool can be used in association with any tool used for *inter-DC* and/or *intra-DC* consolidation. This is a semi-automatic tool as the collection and organization of the input data required for analysis is not a part of the tools functionality.

A. Related Work

Information regarding DC network design is described in [1] and associated infrastructure needs and cost of ownership are described in [2].A network architecture where in the Layer-2 switches are replaced with a set of servers acting as load balancers (Valiant Load Balancing technique) has been proposed in [3]. In [4], Cisco provides a system to validate the design of DC Virtualization and Cloud deployment, which can be used for network design. For instance, VMware [5] deployment on the cloud is performed using a validated design consisting of Cisco Nexus Switches, Cisco Unified Computing system and NetApp storage solution [6]. In [7], a tool to solve infrastructure performance, Quality of Experience (QoE) and Return on Investment (ROI) models is described. This system aims at providing information about infrastructure capacity up-gradation but does not relate to optimization. An infrastructure management tool which alerts about deviation from normal operating conditions in system parameters is explained in [8]. Though this system helps in management of the infrastructure it does not deal with transformation/optimization. An infrastructure monitoring and reporting tool without any transformation/optimization is described in [9]. Automated infrastructure management systems have been described in [10] and [11].

Though there are various proposals in literature for DC consolidation, DC network design and IT infrastructure management, significant contribution on application traffic modeling/prediction, usage analysis, etc., are not explored fully for the network optimization/transformation activities, which we explore through this paper in Section 2. In Section 3, we evaluate the performance of the tool and we conclude this paper in Section 4.

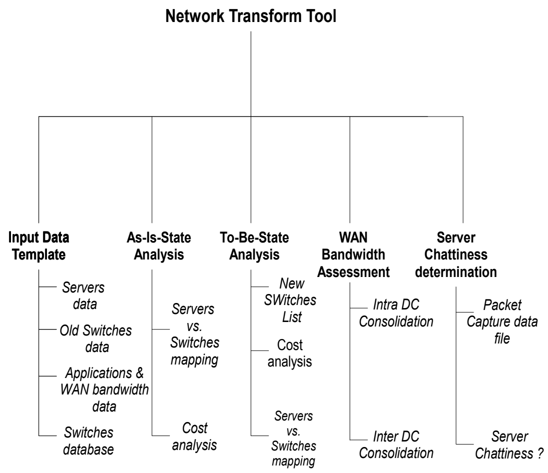

Fig. 1. Components of the tool.

II. Network Transform Tool

The Network Transform Tool is developed using Java JDK 6 [12] and NetBeans IDE [13]. The input data required for analysis performed by the tool and the algorithms used are

Manuscript received: September 14, 2012; revised October 29, 2012.
The authors are with TCS Networks Lab, Bangalore, Karnataka, India (e-mail: {sm.nadaf, hemant.rath, anantha.simha} @tcs.com).

explained in the next sections. The tree diagram in Fig. 1 depicts the components of the tool and Fig. 2 depicts the high-level overview of function of the tool.

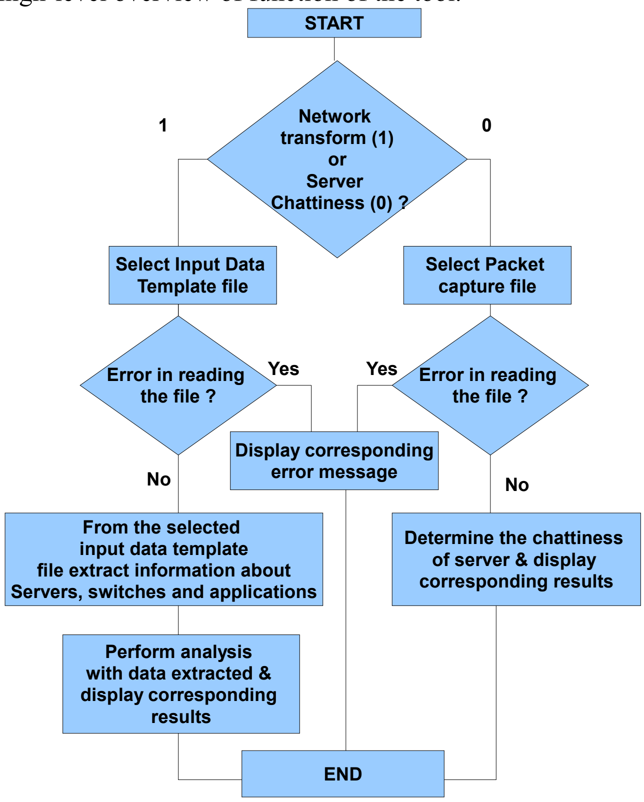

Fig. 2. Flow chart of process followed by the tool.

A. *Input Data Template*

Information about the servers, switches, network architecture, consolidation strategy used (i.e., *inter-DC* or *intra-DC* consolidation collected from IT consolidation team), database of switches offered from different vendors and WAN bandwidth consumed by the applications are present in the input data template. Also the configurable parameters of the tool such as switch vendor selection (i.e., Cisco, Juniper, Brocade etc.), existing switches reuse option, threshold values selection (γ and other parameters), expected values selection (α1, α2 and other parameters) etc., are a part of the input data template. This template is used by the tool for performing the analysis of *As-Is-State* (existing state) and *To-Be-State* (new state) network.

B. *As-Is-State Analysis*

With the information extracted from the input data template, the tool performs analysis of the *As-Is-State* network and formulates the associations mentioned below:-

- Servers vs. Switches mapping for As-Is-State Network: The mapping (i.e., port-to-port connectivity) between the As-Is-State servers and switches.
- Cost analysis for As-Is-State Network: From the switches database in the input data template, tool can extract the information about the throughput and approximate cost associated with a particular configuration of a switch. The tool examines the configuration of each of the existing switches, and then determines the effective throughput and approximate cost incurred for each DC in the As-Is-State Network.

Note – Currently, the Switches database is populated with Cisco's models and their capability. Work in progress to include switch models from other leading networking vendors as well.

Algorithm 1: Network Transformation Process

1. **Start**
2. Compute the effective throughput and approximate cost for each DC in the *As-Is-State* network.
3. **for** *i = 1 to number of data centers* (N_{dc})
4. Determine the number of new and old servers.
5. Determine the ratio of new servers to old servers (r).
6. **if** r <= *Threshold value* (γ)
7. Number of target data centers (N_{tdc}) += *1*
8. **end if**
9. **end for**
10. Compute the number of different type of ports present in each of the target data center based on the port bandwidth.
11. Compute the expected minimum and maximum numbers (α_1 & α_2) for each port type per switch in a target DC.
12. Compute the minimum and maximum number (M_{min} & M_{max}) of all port types supported by the switches.
13. Determine the switches which can provide connectivity to the target data centers using the values of $\alpha 1$ & $\alpha 2$.
14. Compute effective throughput and approximate cost for each target DC.
15. **if** *Effective throughput and Approximate cost satisfy*
16. Go to Step 21
17. **else**
18. Alter values of α_1 & α_2.
19. Go to Step 13
20. **end if**
21. **End.**

C. *To-Be-State Analysis*

For the *To-Be-State* analysis, the tool extracts information about the consolidation performed, network port configuration of new servers and the configuration of existing switches (for reusability check) from the input data template. This tool uses Algorithm 1 which we explain later. The tool formulates the following associations:-

- List of Switches for To-Be-State Network: The tool calculates the network port requirement of the new servers, i.e., it calculates the number of 10/100/1000 Mbps (Megabits per second), 1 Gbps (Gigabits per second), 10 Gbps, 40 Gbps and 100 Gbps ports present. Then, from the switches database present in the input data template the tool finds the set of switches along with the configuration which can provide connectivity to the new servers. The tool can also examine the possibility of reusing some of the existing switches with minor modifications if the user configures this option in the input data template. Work for developing algorithm and code for re-usability check is under consideration.
- Cost Analysis for To-Be-State Network: In this process, the tool determines the effective throughput and approximate cost for each DC of the To-Be-State Network.
- Servers vs. Switches Mapping for To-Be-State Network:

After the switch selection process is completed, the tool generates a mapping that can be used to connect the new servers and the switches of the To-Be-State Network.

Step 2 of the above algorithm computes the effective throughput and approximate cost for *As-Is-State*. Steps 3 to 9 determine the number of target DCs available for transformation. Step 10 determines the number of different port types in a DC based on the port bandwidth and Step 11 computes the expected minimum and maximum number ($\alpha 1$ & $\alpha 2$) of ports per switch in a DC belonging to a particular port type. Steps 12 to 14 determine the list of suitable switches with configurations for *To-Be-State* and also the effective throughput and cost associated with them. Steps 15 to 20 verify whether conditions are satisfied or not. If conditions are not satisfied then switch re-selection takes place else selection is complete

D. WAN Bandwidth Assessment

In this process, the tool determines the effective WAN bandwidth required by each target DC after the consolidation. The tool extracts application WAN bandwidth information from input data template and uses Algorithm 2 for WAN bandwidth assessment which we explain later. The below mentioned cases are observed for the WAN bandwidth assessment:

- Intra-DC Consolidation: The WAN bandwidth required by each target DC after consolidation is equal to WAN bandwidth before consolidation since no servers are moved between the DCs and also DCs themselves are not consolidated to fewer ones.
- Inter-DC Consolidation: The WAN bandwidth required by each target DC after consolidation is equal to sum of WAN bandwidth required by all groups of servers present in that DC. The WAN bandwidth required by a group of servers moved between two DCs is computed by the applications WAN bandwidth information.

Algorithm 2: WAN bandwidth Assessment.

```
1.  Start
2.  Find the type of consolidation followed.
3.  if intra-DC consolidation
4.     Go to Step 13
5.  else
6.     for h = 1 to number of data centers (N_tdc)
7.        for j = 1 to (number of server groups)
8.           Determine the WAN bandwidth required by the
             server group (S_BW(j)).
9.           DC WAN bandwidth (D_BW(h)) += S_BW(j)
10.       end for
11.    end for
12. end if
13. End.
```

Step 2 of the above algorithm finds the type of consolidation performed (i.e., intra-DC or inter-DC consolidation). If intra-DC consolidation is performed then process ends else WAN bandwidth assessment is made. Steps 7 to 10 determine the WAN bandwidth required for a target DC as the sum of WAN bandwidths required by all the server groups present in a target DC. This process is repeated to find the WAN bandwidths required for all target DCs.

Note that, Algorithm 1 is a simple algorithm and has O(N) complexity, where N is the number of DCs, whereas Algorithm 2 has O(NK) complexity, where N is the number of DCs and K is the number of server groups. Since both N and K are relatively smaller numbers (e.g., number of DCs usually consolidated is less than 10 and typical value for server groups is less than 20 per DC), it is easy to implement and evaluate in real-time.

E. Server Chattiness Determination

In this process, the tool determines whether a server/application is chatty in nature or not. We measure the chattiness of a server as the number of data packets exchanged in a set of multiple conversations, where a conversation is the set of packets sent and received between two nodes. The tool reads the packet capture data (.cap or .pcap) file and computes the total number of conversations, source and destination for each conversation and the total number of packets for each conversation. The tool makes use of the threshold value for total packets for a conversation (NP_{th}) to determine whether a conversation is chatty in nature or not and the threshold value for total number of chatty conversations (NC_{th}) to determine whether a server is chatty in nature or not. This information is useful to determine the response time of an application.

III. Performance Evaluation

A. Network Transform Evaluation

We verified the performance of the tool for both *inter-DC* (Case 1) and *intra-DC* (Case 2) transformations. For *inter-DC* transformation we considered consolidation of 4 base DCs to a single target DC with consolidation of servers. Each DC is having 2-tier network architecture with access and aggregation layer switches. The tool performed analysis of *As-Is-State* network consisting of 12 modular switches and found a total effective throughput equal to 2130 Mpps (Million packets per second) and total approximate cost equal to $46,000. After the *To-Be-State* network analysis, tool obtained a 2-tier network architecture map for the target DC consisting of 4 modular configuration switches with total effective throughput equal to 2160 Mpps and total approximate cost equal to $18,000. Also, the tool made an assessment of the WAN bandwidth required for the target DC to be equal to sum of WAN bandwidth associated with the 4 base DCs.

For *intra-DC* transformation we considered consolidation of servers inside each of the 4 base DCs without consolidating any DC. The tool performed analysis of *As-Is-State* network consisting of 29 switches in total and having a combination of both modular and fixed configuration switches and found a total effective throughput equal to 6282 Mpps and total approximate cost equal to $98,500. After the *To-Be-State* network analysis, tool obtained a 3-tier network architecture map for each base DC. The tool suggested a total of 14 switches with both modular and fixed configuration, and having a total effective throughput equal to 8642 Mpps and total approximate cost equal to $64,500. Also, tool indicated no change in the WAN bandwidth required for each base DC since it is an *intra-DC* transformation. Fig. 3 shows the analysis charts generated by the tool for Case 1 and Case 2.

Fig. 4 shows a screenshot of the tool developed. The tool consists of three main tabs: - *As-Is-State*, *To-Be-State* and WAN bandwidth assessment. In the *As-Is-State* tab the tool displays two tables, where the first table provides information about the mapping already existing between old servers and switches; and the second table gives information about effective throughput and approximate cost of old switches. In the *To-Be-State* tab the tool displays three tables. First table suggests list of new switches and their configurations and the second table gives information about the effective throughput and approximate cost of new switches. Third table suggests a mapping for new servers and switches. WAN bandwidth assessment tab displays two tables which give information about WAN bandwidth of *As-Is-State* network and the WAN bandwidth required for the *To-Be-State* network. Apart from the main tabs, the tool also contains a help feature to guide a user in understanding the functions of the tool.

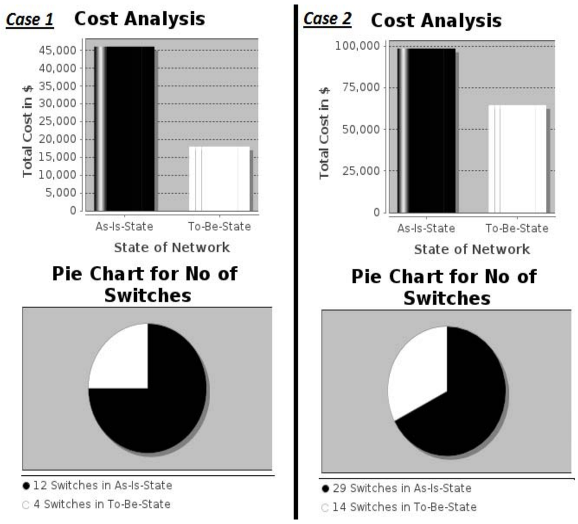

Fig. 3. Analysis charts generated by the tool for case 1 and case 2.

B. Server Chattiness Evaluation

We used two packet capture files (captured for short duration) from Server1 (S1) and Server2 (S2) to verify and validate the chattiness process of the tool. Fig. 5 and 6 depict the analysis chart generated by the tool for S1 and S2 respectively. The threshold values NP_{th} and NC_{th} considered here are 50 and 2 respectively. In case of S1, the tool identified 3 conversations with source IP's A = 192.168.0.1, B = 192.168.0.105 and C = 192.168.0.102 and for S2 the tool identified 4 conversations with source IP's A = 10.132.0.55, B = 10.132.0.40, C = 10.132.0.38 and D = 10.132.0.49. After comparing the total packets per conversation and the number of chatty conversations for S1 and S2 with threshold values NP_{th} and NC_{th}, the tool identified S1 to be chatty in nature and S2 to be not chatty in nature.

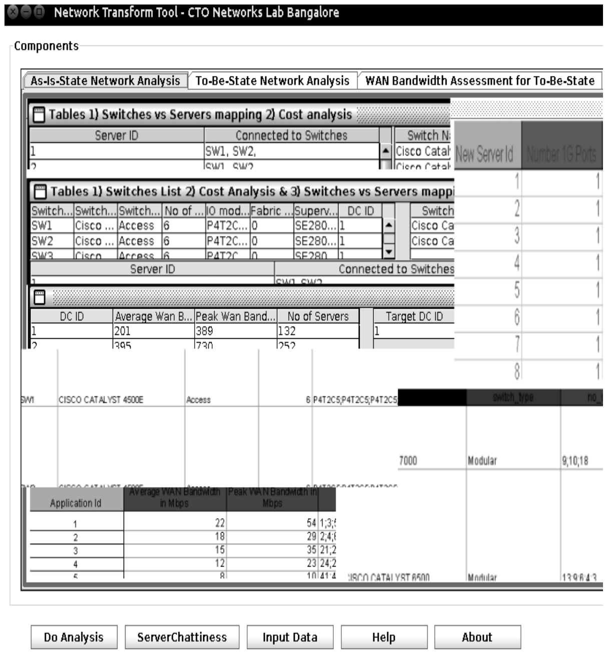

Fig. 4. Screenshot of the tool developed (screenshots of results displayed by the tool and input data given to the tool merged here).

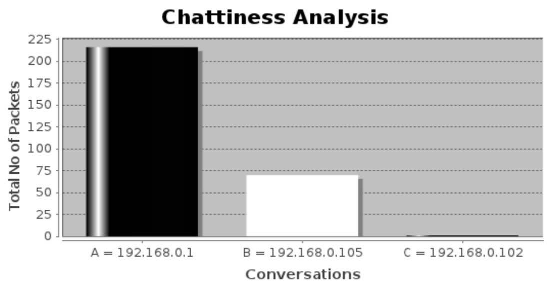

Fig. 5. Charts generated by the tool for Server 1 chattiness determination.

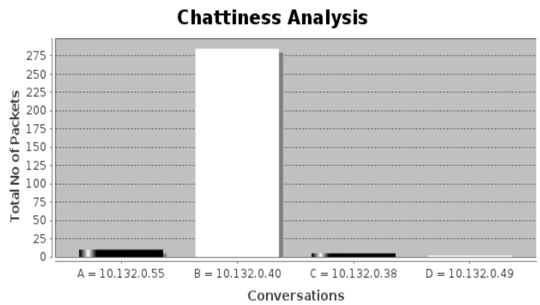

Fig. 6. Charts generated by the tool for server 2 chattiness determination.

IV. Conclusion

In this paper, we describe the components and functionality of our proposed Network Transform Tool and also discuss about the performance of our tool with the help of two experimental input cases of *intra-DC* and *inter-DC* transformations. From the results obtained, it is evident that our tool provides a significant reduction in CAPEX and OPEX related to the network infrastructure. Also, our tool mainly helps to carry out some elementary tasks of network design such as ports calculation for servers, throughput and port requirements matching, network components (i.e., switches) selection, network mapping and cost analysis in a semi-automated manner. This tool is a semi-automatic tool and has the potential for extensive usage required for

Network Transformation. At present we are in the process of deploying this tool for practical cases where network transformation is required along with IT transformation. Based on the feedback obtained from the deployment team, this tool can be improved further and can be made more automatic in future.

REFERENCES

[1] Cisco Data Center Infrastructure 2.5 Design Guide. [Online]. Available: http://www.cisco.com/application/pdf/en/us/guest/netsol/ns107/c649/ccmigration_09186a008073377d.pdf

[2] What goes into a Data Center? [Online]. Available: http://research.microsoft.com/pubs/81782/2009-06-19 Data Center Tutorial - SIGMETRICS.pdf

[3] Towards a Next Generation DC Architecture: Scalability and Commoditization. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=79348

[4] Cisco Validated Design (CVD) for Datacenter. Virtualization and Cloud. [Online]. Available: http://blogs.cisco.com/datacenter/cisco-validated-design-cvd-for-datacenter-virtualization-and-cloud/

[5] VMware – virtualization software. [Online]. Available: http://www.vmware.com

[6] NetApp – Network storage solution [Online]. Available: http://www.netapp.com

[7] M. R. Haley and D. C. Verma, "Converged tool for Simulation and Adaptive operations combining IT Infrastructure Performance, Quality of Experience and Finance parameters," December 27, 2007.

[8] P. V. Ravi, P. K. Rao, W. E. Jensen, and R. A. Smith, "Automated information technology infrastructure management," March 9, 2006.

[9] J. D. Thomas, "Heterogeneous Information Technology (IT) Infrastructure Management Orchestration," February 18, 2010.

[10] Towards An Automatic IT Infrastructure Management Paradigm for Manufactory and Enterprise. [Online]. Available: http://eil.stanford.edu/publications/jie_wang/ICMA2003-Jie.pdf

[11] Architecture for the Automated Management of Data Center IT Infrastructure. [Online]. Available: http://www.mnm-team.org/pub/Dissertationen/grau10/PDF-Version/grau10.pdf

[12] Java Development Kit 6 (JDK 6). [Online]. Available: http://www.oracle.com/technetwork/java/javase/

[13] NetBeans Integrated Development Environment (IDE). [Online]. Available: http://www.netbeans.org/

Shameem holds BTech from Visveshwaraya Technological University (VTU) Belgaum in Electronics and Communication Engineering. Since October 2011, Shameem has been working as a Researcher at TCS Networks Lab, Bangalore, India. He has close to 4 years of experience in Java OSGi based Home/Car gateway development, Nokia smart-m3 platform based Interoperability middle-ware development, Real-time 3GPP LTE Physical layer simulation using Matlab and C in Cloud/Distributed Computing Environment and Bandwidth aggregation of multiple physical links in Linux. His current research interest includes: Social media traffic measurement and analysis, Optimum Base Station placement for Indian terrain, Enterprise Network Transformation and Optimization etc., He has published and presented many research papers in national and international conferences/ seminars such as IEEE ICCAIE, IARIA, UBICOMM, IEEE NCC, etc., and has filled 3 patents through TCS.

Hemant holds MTech and PhD from IIT Bombay, India (both in Communication Engineering) and BE in EL&TCE from VSSUT Burla (formerly UCE Burla), Sambalpur, Odisha. He has close to 15 years of experience in academics, research and industry. Since Dec 2010, Hemant is working as Scientist in TCS Networks Lab, Bangalore. His current research interest includes: QoS in Networks, LTE/WiMAX Scheduling, Self-Optimization, Propagation Model Design, Speech Processing, M2M Communication, Cloud Computing, Modeling of Social Network Traffic, Mobile Network Optimization, etc. He has published many research papers and presented many talks in national and international conferences/seminars such as IEEE Globecom, IEEE ICC, IEEE COMSWARE, IEEE COMSNETS, IEEE WPMC, NCC, ITU-T, BWCI-COAI Workshop etc., and has filled several patents through TCS. He also actively participates in standardization activities (IoT – GISFI, IoT – ITU-T and 3GPP).

Anantha is a post graduate in Electrical Engineering from IIT Madras, India and BE in Digital Electronics from University of Bangalore, India. He has 30+ years of experience in industry in the areas of microprocessor based hardware design, embedded protocol software, enterprise networking. For the last 19+ years he has been working with TCS. Currently he is heading the Networks Lab, Bangalore under CTO organization of TCS. His current research interest includes: QoS in Networks, Scheduling, Self-Optimization, Large Scale Network Design, Cloud Computing etc. He has published many research papers in national and international conferences/seminars and has filled several patents through TCS.