

# Nassim<sup>[1]</sup>阅读报告

黄磊 计 702 2022E8013282156

## 背景介绍

网络是现代社会高效运转的重要基础设施。大型企业往往会从多家供应商处采购和运营网络设备，需要不断将新的设备型号和新的供应商引入他们的网络，而这种多供应商的性质使得网络管理任务变得十分复杂。为了减轻管理多厂商网络的难度，高效地将多种设备吸纳入统一的配置管理框架，是支撑海量网络应用的核心基础能力。网络运营部门一直使用 SDN（Software-Defined Network，软件定义网络）管理方法，它为所有网络设备提供了一个逻辑上集中的控制平面。SDN 的核心数据结构是统一设备模型(Unified Device Model, UDM),它将所有厂商的设备配置模型标准化，结合相关工具，SDN 控制器就可以相同地配置多供应商设备。

UDM为管理现有设备提供便捷，但由于缺少用于配置和操作的标准化接口，每个运营商SDN网络的UDM有可能不同，将新的供应商或者不同配置型号的设备引入网络就会变得十分困难。并且，对于一些传统设备而言，SDN对其的控制只能通过Telnet连接命令行界面（Command Line Interface, CLI）发送指令以实现高级操作，因而这要求SDN控制器能够“理解”命令行语句并在UDM与设备模型之间建立映射。当前设备之间的同化操作(Software-Defined Network Assimilation, SNA)完全依赖网络运营工程师，其根据用户手册手工制作新设备和传统设备的模型，并将模型映射到UDM。整个过程不仅需要大量的人力和精力，并且极其容易出错。

具体而言，SDN的同化操作（即SNA）具备三个主要挑战：

1. 用户手册格式的异构性：用户手册通常是获取配置模型最值得信赖的来源，尤其是对于只有CLI访问权限的传统设备。但手册语法没有标准化，不同的供应商之间手册结构和语法可能存在巨大差异，这使得语料库解析变得十分困难。

[1] Software-Defined Network Assimilation: Bridging the Last Mile Towards Centralized Network Configuration Management with NAssim <https://dl.acm.org/doi/10.1145/3544216.3544244>

Attribute \ Vendor	Huawei	Cisco	Nokia	H3C
CLIs	<class="sectiontitle">Format	<class="pCE_CmdEnv">	<class="SyntaxHeader">Syntax	<class="Command">Syntax
FuncDef	<class="sectiontitle">Function	<class="pB1_Body1">	<class="DescriptionHeader">Description	<class="Command">Description
ParentViews	<class="sectiontitle">Views	<class="pCRCM_CmdRefCmdModes"> Command Modes	<class="ContextHeader">Context	<class="Command">View
ParaDef	<class="sectiontitle">Parameters	<class="pCRSD_CmdRefSynDesc"> Syntax Description	<class="ParametersHeader">Parameters	<class="Command">Parameters
Examples	<class="sectiontitle">Examples	<class="pCRE_CmdRefExample"> Examples	/	<class="Command">Examples

表1 设备用户手册的语法多样性

2. 手册中的错误与歧义：手册中不可避免的存在一些错误和歧义，盲目使用手册内容进行解析会影响SNA的效率与正确性，甚至导致生产问题。而该部分往往需要专家进行判断和纠错。
3. 配置模型之间的异构性：为了与竞争队友进行区分并保护设计专利，不同厂商的模型配置语言会有明显的不同。即便是一些很小的任务，模型配置也会存在巨大差异。将这些不同的配置模型与UDM相匹配是一项挑战。

Intent	Cisco	Huawei	Juniper
check vlan	show <b>vlan</b> [vlanid]	display <b>vlan</b> [vlanid]	show <b>vlan-id/vlans</b> [vlanid]/[vlanname]
add/delete vlan	<b>vlan</b> [vlanid]/no <b>vlan</b> [vlanid]	<b>vlan branch</b> [vlanid]/undo <b>vlan branch</b> [vlanid]	set <b>vlan-id</b> [vlanid]/delete <b>vlan-id</b> [vlanid]
configure spanning tree root bridge	spanning tree <b>vlan</b> [vlanid] root primary	stp <b>instance</b> [vlanid] root primary	spanning-tree <b>vlan-id</b> [vlanid] root primary

Table 2: Configuration syntax comparisons across Cisco, Huawei, and Juniper

表 2 Cisco、Huawei 和 Juniper 的配置语法对比

面向网络基础设施需要吸纳新设备的长期而持续的需求，高效准确获取设备原生配置模型和网络统一配置管理模型的映射关系是一个核心挑战。华为首次研究该任务，并创新地提出了同化概念，推出了助力网络配置管理最后一公里的辅助框架 NAssim，把一个现有网络“消化、理解、吸收”新设备的过程尽可能进行了自动化，大幅提升网络运维的效率。本篇论文是中国地区首篇荣获该会议最佳论文奖的论文。

解决方案

一、 总体思想是什么？

受生物学里同化作用的启发，研究团队提出了 SNA (Software-defined Network Assimilation，软件定义网络同化) 概念以及辅助框架 NAssim，将包括传统设备和新引入设备等异构的设备配置模型以统一模型配置纳入进现有的集中式控制

[1] Software-Defined Network Assimilation: Bridging the Last Mile Towards Centralized Network Configuration Management with NAssim <https://dl.acm.org/doi/10.1145/3544216.3544244>

的网络中，其核心难题是解决多样的设备原生配置模型（Vendor-specific Device Model, VDM）和中心控制器统一配置管理模型（Unified Device Model, UDM）的映射问题。因此，根据上述分析的难点，将 SNA 分为两个阶段。第一阶段为设备配置模型构建阶段，即 VDM 构建阶段，NAssim 帮助运维工程师从手册中提取并验证 VDM；第二阶段为映射过程，NAssim 将 VDM 参数映射到 UDM。

各个模块/部分如何设计？为什么？

针对上述分析的三个挑战，NAssim 在两个阶段内使用了三个核心组件进行应对和解决，分别是解析器、校验器和映射器。

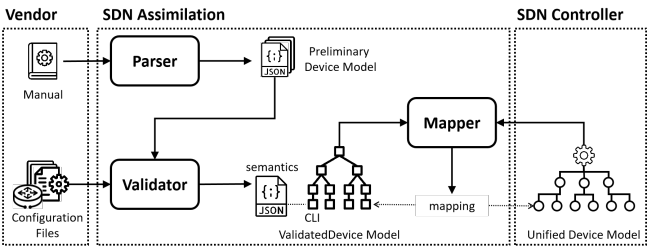


图 1 NAssim 总体结构

### 2.1 阶段一：设备配置模型（VDM）构建阶段

在阶段一中，NAssim 的目标是基于配置手册构建出经改进和验证的精确可靠的设备配置模型，该阶段设计两个核心组件，分别为解析器和校验器。在该阶段，NAssim 使用树形结构刻画设备配置模型，包含设备所支持的配置集合、层级关系以及语义信息，树的节点代表一个配置模板，并分别连接到对应的配置手册中的语义资料，每条边代表模板之间的关系。

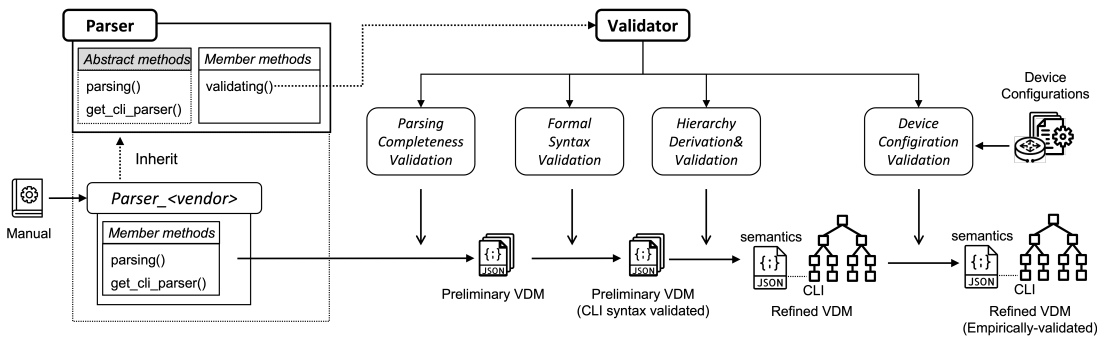


图 2 VDM 配置阶段框架中解析器和验证器的工作流程

[1] Software-Defined Network Assimilation: Bridging the Last Mile Towards Centralized Network Configuration Management with NAssim <https://dl.acm.org/doi/10.1145/3544216.3544244>

### 2.1.1 解析器

解析器框架的主要目标是从手册中提取所有与 VDM 相关的基本信息，并将不同风格的手册标准化为统一格式以便于后续处理。基于对已有配置信息的调研，核心的配置信息为设备所支持的所有配置命令及其功能、配置命令的工作视图、配置参数及其含义描述和配置样例。考虑到表达性、可解释性和可扩展性，NAssim 捕捉了来自不同供应商手册的共性，并采用 JSON 文件格式化配置信息，得到了独立于供应商的信息配置格式。

Keys	Type Restriction
CLIs	a list of string (non-empty list)
FuncDef	string
ParentViews	a list of string (non-empty list)
ParaDef	a list of dict (Keys: "Paras" and "Info")
Examples	a list of list

**Table 3:** Format Definition of Vendor-Independent Corpus (JSON)

表 3 独立于供应商的配置信息格式

为了保证解析的完整性，NAssim 的解析框架采用测试驱动开发（Test-Driven Development, TDD）原则来指定构建解析器的工作流。在具体实现中，解析器充当解析器基类，并由其子类 `Parse_<vendor>` 继承，如图 2 上游所示。原则上，我们只需要在上游人工为每个供应商实现一次 `Parse_<vendor>`，并调用验证器进行验证，这个框架便可以将一个统一的、可扩展的测试方案容纳到基本解析器类中，以使它的所有子类受益，尽量减少解析错误造成配置信息的损失，保证后续工作顺利进行。

### 2.1.2 验证器

解析器框架引入了 TDD 原则，大大减少了由于粗心解析造成的信息丢失。但用户手册终究是人工书写的文档，难免有错误和歧义，而这部分无法在解析器中被检测出来。因此校验器的主要目标是校验配置解析初步配置语料从而生成精

确可靠的设备配置模型，并在命令级、命令互关系级和命令片段级三重校验保证正确性。

在命令级层面，验证器会进行形式语法验证，在解析的配置语料库中审核校验 CLI 命令。每本配置手册的序言部分通常会对手册配置模版语法进行说明，例如许多配置手册都采用花括号代表选择分支，中括号代表可选分支。在验证过程中，先将这些表达等效转换为 BNF（Backus Normal Form）形式，并通过解析生成工具得到语法解析器，该语法解析器会对配置解析语料中的配置命令模版进行自动化校验，识别有问题的字段交予运维专家进行审阅和纠正，从而提高 VDM 语料库的可信度。

Command descriptions use these conventions:

Convention	Description
<b>boldface font</b>	Commands and keywords are in boldface.
<i>italic font</i>	Arguments for which you supply values are in italics.
[ ]	Elements in square brackets are optional.
{ x   y   z }	Alternative keywords are grouped in braces and separated by vertical bars.
[ x   y   z ]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

表 4 Cisco 用户手册中的命令行惯例

```
import pyarsing as p

# syntax parser for Cisco CLI
word = p.Word(p.printables, exclude_chars='{}[]|#\n').setParseAction(leaf_gen)
ele = p.Forward()
items = ele + p.ZeroOrMore(' ' + ele)
select = p.Group('{ ' + items + ' }').setParseAction(select_gen)
option = p.Group('[ ' + items + ' ]').setParseAction(option_gen)
ele <|= p.OneOrMore(option ^ select ^ word).setParseAction(ele_gen)

syntax_parser = ele
```

图 3 语法解析器生成的代码片段

在命令互关系（也叫模型层次结构）级层面，验证器利用配置手册的配置样例推理出配置命令间的层级关系。在图 2 的 Refined VDM 中，设备配置模型的命令通常为基于树的层级结构，虽然大部分配置手册未显式展现，但由于配置样例通常是当前配置命令模版的实例化片段，片段中也包含当前配置命令的父命令，因此通过配置实例和配置模版的匹配，可以推导出配置命令模版之间的层级关系。该过程通过 CLI 图模型方法（CLI Graph Models, CGM）和 CLI 模板匹配算法（CLI Instance-Template Matching, CITM）实现。CGM 是一个具有单个根和单个终端的有限状态机，用于确定 CLI 实例是否与 CLI 命令模板匹配。CITM 使用广

[1] Software-Defined Network Assimilation: Bridging the Last Mile Towards Centralized Network Configuration Management with NAssim <https://dl.acm.org/doi/10.1145/3544216.3544244>



度优先搜索找到从根节点到接收器的路径，并在每次搜索步骤中找到下一个元素的所有潜在候选元素进行匹配验证。

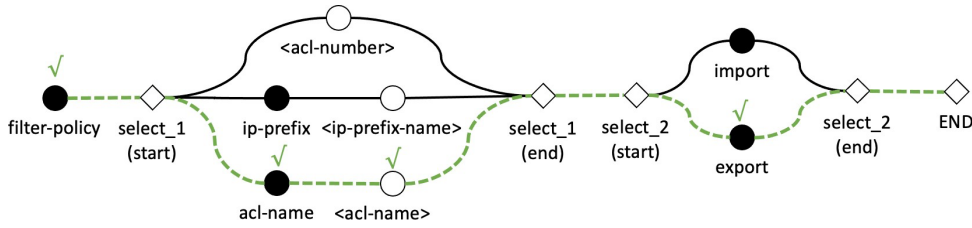


图 4 CLI 实例与 CLI 模板的图形模型匹配

CLI 模型层次结构派生方案可以恢复大部分层次结构，但不能保证完全恢复，因为原始手册中可能缺少数据。因此，研究者在 CLI 层次结构派生之后引入了一个额外的验证步骤。即在命令片段级，验证器进一步利用经验配置文件，对设备配置模型进行校验。对于配置文件中的每个 CLI 实例，我们首先找到其匹配的 CLI 模板；然后，我们检查匹配的模板及其父 CLI 实例的模板是否在 CLI 层次结构上形成父子关系。我们记录不匹配的 CLI 实例和原因(例如，未找到匹配的 CLI 模板、不匹配的层次结构)，提交给专家进行审核。

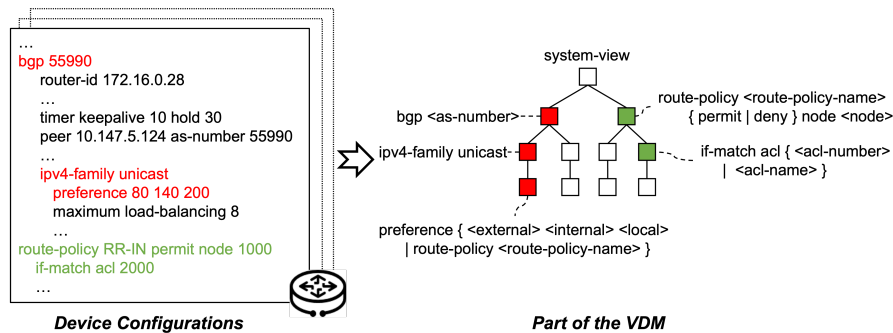


图 5 根据经验设备配置进行验证

对于那些在经验配置中未使用的 CLI 命令，我们利用 CGM 来生成 CLI 命令实例；然后将实例直接发送给设备进行验证；最后，我们使用特定命令（检查实例是否在设备中正确配置。若正确配置，则该实例成为经验实例。再重新运行上面的工作流以完成校验。

## 2.2 阶段二：配置模型映射（VDM-UDM）阶段

在阶段一中，解析器与校验器生成了精准的设备配置模型（VDM），而要将这些配置模型同化到控制器的统一配置模型（UDM）中，则需要做 VDM-UDM 映射，即第二阶段的映射阶段。

第二阶段核心是维护一个映射器，利用自然语言处理领域的最新技术，设计并训练强大的配置语义理解模型，基于两个异构配置模型上的配置语义信息，进行自动化推理和映射，主要是通过对预训练模型 SBERT 进行网络配置领域适应后得到的配置语义理解 AI 模型 NetBERT。其工作流程主要分为配置语义上下文提取，配置语义编码，和配置语义相似度计算和映射三部分，最后还会对网络进行微调处理。

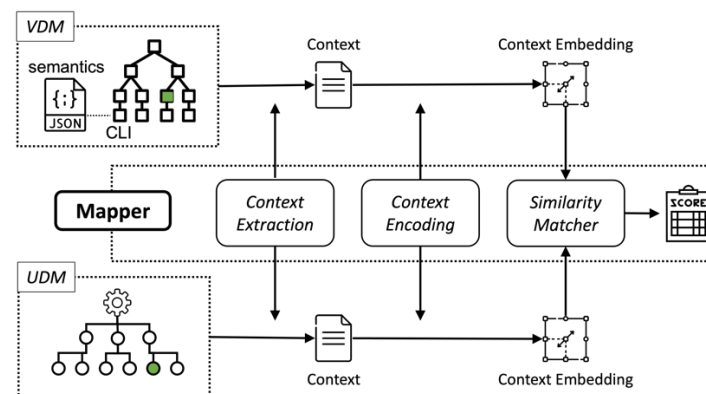


图6 映射器的细节工作流程

在上下文提取部分，对于每个参数，从 VDM（或 UDM）中提取相关的语义信息作为其上下文。实验中发现参数名和 CLI 命令名、参数的描述、父视图以及 CLI 命令的功能描述这些信息对于参数映射帮助很大。不同配置模型可以提取不同数量的文本信息作为配置语义上下文。

在配置语义编码部分，将经过预训练的 SBERT 模型进行适应性改造得到 NetBERT 网络，对不同配置模型提取不同数量的配置语义，并对每个信息进行单独编码，并输出其在隐空间的嵌入表征向量矩阵。

在配置语义相似度计算和映射部分，映射器会计算来自不同配置模型的一对参数对应的嵌入向量的余弦相似度来表征其配置语义的相似度，并对打分进行排序，选择前 k 个相似参数进行映射推荐。

此外，运维专家会给定几个手动标记的映射对以对 NetBERT 进行微调。将标记的映射对视作正对，随机采样生成负对。对每一对参数，提取上下文，并输入到当前的 NetBERT 模型当中，使用与原始 SBERT 模型完全相同的架构模型和句子匹配训练目标来进行微调。

## 性能评估

使用 Python 3.8 实现了 NAssim 的原型，并分别对两个阶段进行评估，分别验证第一阶段的可靠性和第二阶段的性能。

### 3.1 VDM 构建的可靠性验证

基于 Huawei、Cisco、Nokia 和 H3C 四个主流设备商的配置手册，使用 NAssim 解析器和校验器构建设备原生配置模型以验证 VDM 构建阶段的可靠性，并从构建时间、纠察错误/歧义数量、匹配度等多个方面衡量，具体实验结果如下：

Vendor/Model/ReleaseYear	-	Huawei/NE40E/2021	Cisco/Nexus5500/2011	Nokia/7750SR/2021	H3C/S3600/2009
Main Statistics	#CLI Commands	12874	278	14046	759
	#Views	607	27	3832	28
	#CLI-View Pairs	36274	366	22734	851
Adaption Cost	parsing() LOC	45	52	57	41
	get_cli_parser() LOC	8	6	10	8
Syntax Validation	#Invalid CLI Commands	13	19	139	13
Model Hierarchy Derivation & Validation	#Example Snippets	15466	523	/	1147
	Construction Time (second)	785.58	14.29	94.56*	34.3
	#Ambiguous Views	47	8	/	4
Device Configuration Validation	#Config Files	197	/	416	/
	Matching Ratio	100%	/	100%	/

表 5 VDM 构建阶段的性能评估

从表中可以看出，其能够检查出超出 200 个错误（Invalid CLI Commands）或者歧义（Ambiguous Views）。在对示例片段开始 CLI 实例模板匹配之前，构建最大的模型层次大约需要 13 分钟，有大约 84% 的处理时间花费在所有 CLI 命令的 CLI 图模型(CGM)生成上。由于每个手册只需要做一次上述步骤，因此该时间

[1] Software-Defined Network Assimilation: Bridging the Last Mile Towards Centralized Network Configuration Management with NAssim <https://dl.acm.org/doi/10.1145/3544216.3544244>



开销是合理的。经验数据评估只在 Huawei 和 Nokia 进行，是因为我们可以从其数据中心网络中获取大量配置文件，比较具有参考性。100%的匹配率也证实层次结构推导的完整性。总体而言，VDM 构建阶段是非常可靠的。

### 3.2 VDM-UDM 映射性能验证

在第二阶段中，研究者以 top k 召回率为评估指标，即正确的映射对在前 k 个推荐项中出现的百分比，k 值越小说明召回率越高，代表推荐的配置映射效果越好。实验中所采取的模式说明如下：

IR：基于信息检索（Information Retrieval）的映射方式。文中选取了最为经典的 TF-IDF 方法；

SBERT：使用三元组网络结构作增强的预训练 BERT 模型；

SimCSE：利用对比学习目标来规范预先训练的嵌入，以提升句子相似性测量任务性能；

IR+DL：IR 方法和 DL 的集成模型，其中 DL 可以是任何基于 DL 的句子匹配模型。IR 方法用于初步选择，DL 用于精细排序。

得到实验结果如下所示：

Mapping Setting	Models	k in recall@top k (%)							
		1	3	5	7	9	10	20	30
Huawei-UDM	IR	41	61	69	76	79	80	90	93
	SimCSE	40	59	66	68	70	72	77	81
	SBERT	53	72	79	81	84	85	89	92
	IR+SimCSE	43	68	75	79	81	82	89	92
	IR+SBERT	56	75	81	85	87	88	91	94
	NetBERT	57	74	80	85	86	87	91	94
	IR+NetBERT	<b>58</b>	<b>78</b>	<b>83</b>	<b>86</b>	<b>88</b>	<b>89</b>	<b>93</b>	<b>95</b>
Nokia-UDM	IR	24	41	48	57	59	60	66	70
	SimCSE	20	31	37	39	39	42	45	48
	SBERT	34	38	49	49	52	52	58	53
	IR+SimCSE	24	35	42	46	48	48	57	61
	IR+SBERT	34	42	52	54	55	58	62	72
	NetBERT	34	43	53	<b>66</b>	67	<b>70</b>	71	73
	IR+NetBERT	<b>35</b>	<b>47</b>	<b>55</b>	65	<b>68</b>	68	<b>71</b>	<b>73</b>

表 6 映射器性能评估

[1] Software-Defined Network Assimilation: Bridging the Last Mile Towards Centralized Network Configuration Management with NAssim <https://dl.acm.org/doi/10.1145/3544216.3544244>

实验结果表明，结合 IR 方法初筛后的 NetBERT 模型映射性能最佳。例如，在对华为设备进行映射的场景中，top9 的召回率为 88%，说明若果允许推荐 9 个映射，运维专家只有 12% 的概率需要查阅配置手册，大幅提升相关人员的工作效率。

## 读后感

作者敏锐地观察到了网络配置任务的痛点：将不同供应商的网络配置纳入到统一控制网路中，往往需要投入大量的人力和精力去阅读和理解手册，并手动进行格式统一。作者开创性地研发出 NAssim 框架以加速 SNA 进程，解放大量的网络工程师等专家人力消耗。在工程实现上，两个阶段、三个组件的架构组成精准且可靠地实现了 VDM 的构建和对 UDM 的映射。此外，作者还在细节上使用多重验证以及利用正负样本对模型进行微调，以保证配置构建和映射的性能。

市面上常用的网络配置模型有 CLI 和 YANG/NETCONF 两种，前者应用广泛，但较为简单；后者更加复杂，但也在逐步推广。作者重点分析了 CLI 网络配置模型的解决方案，并在“讨论”部分中，简要解释了 NAssim 也可用于解决 YANG/NETCONF 网络配置，但具体数据仍需进一步研究和补充。

我非常佩服作者敏锐的洞察力和严谨创新的科研作风，其能发现基础领域中的研究空白，并致力于改造它、突破它，跨越关键领域的最后一公里。此外，作者对问题分析全面且细致，解决思路清晰有条理，能充分考虑到各种实现上的细节，并通过“验证”和“微调”的方式增加系统可靠性和容错性，这种周全、严谨、细致的科学研究精神值得我们学习。

## 作者及实验室介绍

### 5.1 第一作者介绍

陈煌枸主页: <https://amyworkspace.github.io/hxchen/>

文章的第一作者是华为香港研究中心 2012 理论实验室的陈煌枸，于 2020 年 8 月在香港科技大学获得了计算机科学与工程博士学位，导师为张倩教授。读博期间，陈煌枸主要从事智能物联网传感和安全交叉领域的课题，之后在华为从事网络配置管理及其与机器学习的交叉研究。陈煌枸对具有实际影响的挑战性问题非常感兴趣，特别关注网络和网络物理系统，其正在进行的研究涉及智能传感、网络物理安全、物联网、网络配置管理及其与机器学习技术的交叉领域。

### 5.2 实验室介绍

华为 2012 理论实验室由任正非与 2011 年成立，是华为的总研究组织，主要负责华为前瞻性技术的预研和开发，其前身是成立于 1996 年的华为中央研究院。其子部门有华为中央研究院、华为中央软件院、华为中央硬件工程院、华为中央媒体技术院、华为产品数字化与 IT 装备部、华为可信理论与技术工程实验室，华为研发能力中心。华为 2012 理论实验室主要面向未来五到十年发展方向，聚焦于基础研究，致力于新一代通信、云计算、音频视频分析、数据挖掘、机器学习等领域的基础层面的创新与突破，是华为最神秘、研发投入最多的实验室。

实验室主页链接暂无。