Manisha Luthra. *Doctoral Symposium: Adapting to Dynamic User Environments in Complex Event Processing System using Transitions.* To be published in the Proceedings of 12th ACM International International Conference on Distributed and Event-based Systems (DEBS), June 2018, ISBN 978-1-4503-5782-1/18/06.

Doctoral Symposium: Adapting to Dynamic User Environments in Complex Event Processing System using Transitions

Manisha Luthra
Technical University of Darmstadt
Germany
manisha.luthra@kom.tu-darmstadt.de

ABSTRACT

Complex Event Processing (CEP) system enables extraction of higher-level information from real-time data streams produced by distributed sources. However, these systems are subject to changes in the user environment e.g., density of sources, rate at which events occur and mobile sources. Therefore, it becomes difficult to satisfy stringent performance requirements posed in terms of Quality of Service (QoS) demands under such a dynamic environment.

This work investigates adaptive use of CEP mechanisms e.g., *operator placement* and *operator migration* by supporting *transitions* i.e., dynamic exchange of these mechanisms. In particular, we build a *transition*-capable CEP system — Tcep to enable integration of multiple heterogeneous CEP mechanisms and allow *cost-efficient* and *seamless* transitions between them. As a proof-of-concept, we have recently designed and developed an initial architecture named Tcep, where we have shown benefits of *transitions* among operator placement mechanisms. In an ongoing research, we explore other CEP mechanisms e.g., *operator migration* and investigate whether *transitions* can bring performance benefits, under the execution of different strategies. In the future, we will investigate if mechanism *transitions* in CEP are beneficial in middleware infrastructures including information-centric networks.

CCS CONCEPTS

• **Computer systems organization → Distributed architectures**; • **Networks → Network dynamics**;

KEYWORDS

Complex Event Processing; Stream Processing; Operator Placement; Migration; Adaptation; Transitions; QoS; Internet of Things

ACM Reference format:
Manisha Luthra. 2018. Doctoral Symposium: Adapting to Dynamic User Environments in Complex Event Processing System using Transitions. In *Proceedings of The 12th ACM International Conference on Distributed and Event-based Systems, Hamilton, New Zealand, June 25–29, 2018 (DEBS '18),* 4 pages.
DOI: 10.1145/3210284.3226051

1 INTRODUCTION

Complex Event Processing (CEP) is an important paradigm to capture, analyze and derive higher-level events from lower-level data streams produced in real-time by distributed data sources. The representation of the higher-level events in a CEP system is done by means of functional computing units called *operators*, such as filters, aggregators, etc.

The Internet of Things (IoT) is one prominent domain in which applications can highly benefit from CEP. For example, authors in work [19] and [10] explore benefits of CEP for low latency critical situations in the context of IoT scenarios such as traffic management and intrusion-detection systems, respectively; Schilling et. al [29] investigates IoT applications of CEP in an industrial context supporting interoperability and heterogeneity; while another authors [5] propose a privacy-driven CEP system for IoT applications. A key property of CEP systems common in all the above approaches is to fulfill Quality of Service (QoS) demands specific to the IoT applications or the end users. For example, QoS demands such as low latency [10, 19], heterogeneity [29] and high trust and privacy [5].

A CEP system complies with such QoS demands by means of *intelligent* mechanisms such as Operator Placement (OP). An OP mechanism defines an embedding of the CEP operators in the form of an *operator graph* to the available IoT resources. These resources may comprise of anything from the fixed cloud data centers to the mobile edge user devices. However, when a user device moves from one location to another, the placement of an *operator* becomes inefficient. For this reason, CEP systems introduce the mechanisms for *operator migration* [12, 20, 21] and *placement update* [22] that allow transfer of an operator between the available resources. Yet, a successful deployment of a CEP system over the IoT resources has to deal with two *major* challenges namely, (i) a highly dynamic user environment and (ii) presence of heterogeneous IoT resources.

To deal with the above challenges, relevant CEP mechanisms must be selected with respect to the environmental conditions and the desired QoS demands. This raises an important problem of CEP *mechanism selection* and *transition* i.e., exchange of CEP mechanisms at run time, which is addressed in this PhD work.

To this end, we design a transition-capable CEP system — Tcep (cf. Figure 2) that is capable of defining transitions between *multiple heterogeneous* CEP mechanisms to fulfill desired QoS demands subject to dynamic user environments. The idea of mechanism transitions origin from the collaborative research center MAKI [23], where researchers investigate mechanism transitions for the communication in Future Internet. Transitions have been investigated in a wide range of communication system mechanisms e.g., event

---

DEBS '18, Hamilton, New Zealand

DOI: 10.1145/3210284.3226051

Manisha Luthra. *Doctoral Symposium: Adapting to Dynamic User Environments in Complex Event Processing System using Transitions.* To be published in the Proceedings of 12th ACM International International Conference on Distributed and Event-based Systems (DEBS), June 2018, ISBN 978-1-4503-5782-1/18/06.

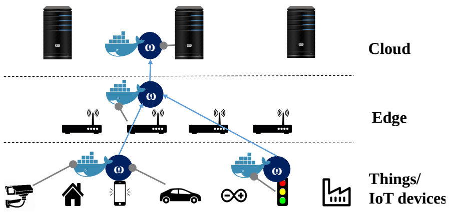

Figure 1: Basic system model for Tcep.

dissemination [25] and filter schemes [24] in publish-subscribe systems; search overlays [6] for peer-to-peer systems and monitoring schemes [27] for service-oriented systems. Our contribution in this work is to design, explore and understand transitions in the context of a CEP system; and to support highly stateful mechanism transitions comprising of many distributed entities.

2 RESEARCH GOALS

We define the following research goals for this doctoral work, to reach our aim of developing a fully transition-capable CEP system.

RG 1: Identification of potential mechanisms of CEP system that can benefit from transitions. In the first step, we identify relevant mechanisms of CEP systems that can benefit from *transitions*. We look into IoT scenarios that face challenges because of dynamic changes in the environment. Consequently, we analyze following research questions. (i) *What are the key mechanisms in CEP that suffer due to dynamic environmental conditions?* (ii) *Which are the scenarios affected by the changes in the environment?*

RG 2: An independent specification of mechanism transition for a CEP system. In this goal, we model transitions and depict the set of environmental conditions that have an influence on the underlying mechanism. The design should be such that, the system is able to adapt to new environmental conditions included at a later point of time. In addition, the mechanism transition concept must be independent of the underlying mechanisms introduced into the system. We define the following research questions. (i) *How to independently specify mechanism transitions for a CEP system?* (ii) *How to quantify influence of the environmental conditions on the CEP mechanisms?*

RG 3: Realization of mechanism transitions for a CEP system. We perform transitions to fulfill desired QoS demands posed by changes in the environmental conditions. However, transitions themselves might impose a cost that needs to be accounted such that, the QoS demands are fulfilled, while maintaining the system integrity. Therefore, we investigate three important questions. (i) *How to realize transitions in a seamless and proactive manner?* (ii) *When to perform a transition?* (iii) *How to select the next mechanism as a consequence of a transition?*

RG 4: Lightweight and cost-efficient execution of mechanism transitions. Another key challenge is the heterogeneity of the underlying infrastructure. This poses a requirement for our system to be platform independent, lightweight and cost-efficient at the same time. As a result, we analyze the following. (i) *How to measure the cost of mechanism transition?* (ii) *How to design mechanism transition concept for heterogeneous infrastructure?*

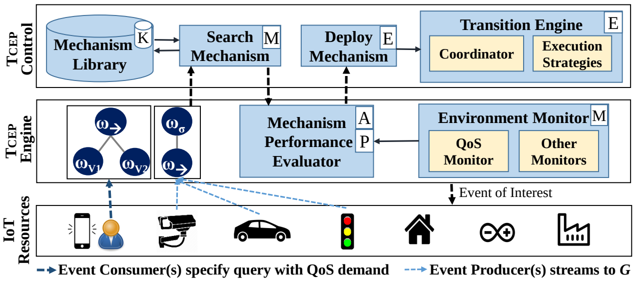

Figure 2: The TCEP system design.

RG 5: Cross layer fulfillment of QoS demands. It is difficult to satisfy the QoS demands based on just the monitored properties of the available physical resources or hosts. In this goal, we investigate cross layer mechanisms e.g., rate control, caching in information-centric networks [33], in order to satisfy the specified QoS demands. Thus, we investigate the following. *How can we take advantage of middleware infrastructures to satisfy stringent QoS demands?*

3 CONTRIBUTIONS SO FAR

We present a basic system model for Tcep in Section 3.1 that will aid us in describing our approach presented in Section 3.2.

3.1 Basic System Model

Tcep consists of a set of event producers (P), which generate continuous data streams (D), a set of event consumers (C), which express an interest in the inference of event patterns from the incoming data streams, and a set of event brokers (B), which host a set of operators (Ω) to process and forward events. Event consumer specifies interest in the detection of an event pattern by means of a continuous CEP query. The query induces a directed acyclic *operator graph* $G = (\Omega \cup P \cup C, D)$, where each vertex corresponds to an operator $\omega \in \Omega$ and each edge corresponds to the processing flow of data streams, s.t., $D \subseteq (P \cup \Omega) \times (C \cup \Omega)$.

Tcep considers resources common in the context of IoT scenarios for the representation of event consumers; producers and brokers (cf. Figure 1). The resources are organized hierarchically consisting of three layers: (mobile) Things referring to IoT devices, a fixed network layer comprising distributed resources in data centers (cloud), and a third layer of resources at the edge consisting user devices e.g., smartphones.

3.2 Tcep: A Transition-Capable CEP System

We aim to design a *generic* transition-capable CEP system that allows an integration of *multiple* and *heterogeneous* mechanisms. The goal is to support *transition* in the system such that it is able to dynamically exchange *any* possible CEP mechanism. Additionally, to support analysis of potentially *any* mechanism. Towards this, we present a preliminary design of Tcep system in Figure 2.

Tcep system comprises of three layers. *IoT resources* including event consumers, at the bottom layer, can pose queries with specific QoS demands. The application components, e.g. the things, will receive in turn the events of interest. The *control layer* utilizes and manages a library of the state-of-the-art CEP mechanisms. It decides when to perform a transition and which mechanism to select within a transition. It is also responsible to coordinate the transition. The *Tcep engine* provides the execution environment for the operators on the IoT resource infrastructure.

Tcep follows the well-known *MAPE-K* [11] model for adaptation. The four processes of the loop, *Monitoring* (M), *Analyzing* (A), *Planning* (P) and *Executing* (E) are realized in a decentralized manner (cf. Figure 2) in the control layer and within the Tcep engine.

Towards our ultimate goal of a generic transition-capable CEP system, we have presented the Tcep design and demonstrated benefit of *transitions* in *Operator Placement* mechanisms (cf. RG 1) [16, 17]. We have shown using an IoT case study, that *transition* between placement mechanisms are essential to deal with dynamic user environment (cf. RG 2 and 3). We evaluated our transition-capable system against fixed placement to show that our system can fulfill changing QoS demands subject to the dynamic user environments. We presented two transition strategies that perform a transition in a *seamless* and *cost-efficient* manner (cf. RG 3 and 4). In order to perform a transition, we presented an adaptive selection method that selects the next mechanism (cf. RG 3).

In the future, we plan to extend our *reactive* adaptation methods to *proactive* adaptation methods and consider other mechanisms of CEP for transition as described in the next section.

4 PLANNED CONTRIBUTIONS

Since this PhD work is in its early phase (start of second year), not all of the research goals (cf. Section 2) have been fully attained. In the following, we define our preliminary plan for each research goal.

RG 1. In [17], we developed and integrated state-of-the-art operator placement mechanisms into the mechanism library (cf. Figure 1). We plan to extend the library by considering other relevant CEP mechanisms for transition e.g., operator migration. Towards this, we will analyze existing migration strategies [12, 20, 21] and their trade-offs to find out which strategy is suitable for the respective environment. In addition, we will analyze the impact of different migration strategies including those presented in [17], on the performance of the Tcep in terms of QoS demands.

RG 2. It is important to know that many different environmental conditions e.g., changes in workload, mobility, topology, etc., have an influence on the underlying mechanisms. It is difficult to quantify the influence of different environmental conditions on the performance of the system. We will be investigating learning approaches to derive performance influence of the different environmental factors.

RG 3. In some situations, the execution time of a transition can be very long and time consuming. In such situations, the reactive methods for adaptation might not be useful anymore. For this, we will develop and analyze *proactive* adaptation methods [18] that will allow us to trigger a transition well-before.

RG 4. In [17], we have analyzed the cost of transition considering two key factors (i) time and (ii) overhead for a single query. We have seen that the cost varies from operator to operator. In this goal, we plan to analyze the cost of transition for other CEP operators e.g., sequence as well as consider other performance factors besides time and overhead.

For better interoperability, we have based our system on a lightweight and portable Docker container similar to other prominent approaches [14]. Yet, none of the existing works have considered evaluation on resource constrained edge devices, as we plan to do.

RG 5. Last but not the least, it is very challenging to guarantee QoS demands, while only considering the physical parameters e.g., hosts. We will consider cross-layer approaches e.g., by means of middleware infrastructures such as information-centric networks [33] to fulfill the specified QoS demands.

5 RELATED WORK

By enabling transitions, Tcep allows to exchange key mechanisms of CEP e.g., operator placement and migration, and in this way, fulfill QoS demands under dynamic environmental conditions. In this section, we compare to the related work in three key areas, (i) operator placement and migration, (ii) adaptive event processing systems and (iii) existing approaches for mechanism transitions.

Operator Placement and Migration. Operator placement (OP) is widely studied to fulfill QoS demands while incurring minimum cost in terms of performance [15]. A wide range of OP mechanisms have been proposed considering different QoS demands such as to achieve low latency [1, 10, 19], to minimize bandwidth [22, 28, 29], to lower message overhead [30], as well as to preserve trust and privacy [5].

The fulfillment of QoS demands, however, is only feasible under limited changes of the environmental conditions. For instance, most existing work [1, 3, 28] builds on stationary networks. Approaches considering dynamic changes, e.g. in the cause of mobility, introduce i) redundancy by means of duplication [4, 30] or checkpointing [13], ii) placement update at regular intervals [22], or, iii) operator migrations when changing the placement [12, 20, 21].

Overall, it is important to note that current approaches for CEP, so far build on a *single* placement or migration mechanism. In contrast, Tcep enables to benefit from the adaptive use of *multiple* existing CEP mechanisms by supporting transitions, while increasing the range at which a CEP system can adapt to meet a specific QoS demand.

Adaptive Event Processing Systems. In this section, we review approaches that have so far considered the adaptive exchange of mechanisms in the context of event processing systems. For example, Weisenburger et. al [32] proposed AdaptiveCEP, a programming model and CEP system that supports specifying QoS demands at run time. This work is complementary to Tcep since AdaptiveCEP is not focusing on the adaptive selection and execution of transition strategies. However, in Tcep the query language is used to specify changes in the QoS demands in order to instantiate a transition.

Heinze et al. proposed an elastic data stream processing system (DSPS) [8], where the number of active hosts can be scaled up and down and operator migration is coordinated accordingly. Based on this work, the same authors proposed an adaptive replication scheme for DSPS [9] that performs adaptation at runtime between active replication and upstream backup schemes for fault tolerance.

Another line of work by Matteis et al. [18] addressed the problem of elasticity by *Model Predictive Control* method that accounts for system behavior over a future time horizon to predict the best reconfiguration to be executed. Aniello et al. [2] proposed an adaptive online scheduling algorithm for Apache Storm using two placement mechanisms. Sutherland et al. [31] developed an adaptive scheduling selection framework for continuous queries in DSPS.

Although the aforementioned approaches benefit from integrating multiple mechanisms, the adaptation between the mechanisms is heavily dependent on the internals of the specific mechanisms in use. Therefore, integrating new alternative mechanisms is a complex task. By offering the abstraction of a transition, Tcep is highly extensible and can easily integrate new mechanisms. Furthermore, no previous work up today has studied the idea of adapting between distinct OP mechanisms.

Mechanism Transitions. Within MAKI [23], mechanism transitions are investigated in the context of a wide range of communication mechanisms [7, 24–27]. Our work builds on and extends the concept of transitions proposed in prior work [6, 24]. By focusing on transitions for OP mechanism, our contribution is the design and understanding of transition for CEP system. We design strategies that can support highly dynamic environment by means of stateful mechanism transitions comprising many dependent distributed entities.

ACKNOWLEDGEMENTS

The author would like to thank her supervisors Prof. Dr-Ing. Ralf Steinmetz and Dr. Boris Koldehofe for their valuable advises and discussions. This work has been co-funded by the German Research Foundation (DFG) as part of the project C2 within the Collaborative Research Center (CRC) 1053 – MAKI.

REFERENCES

[1] Y. Ahmad and U. Çetintemel. Network-aware query processing for stream-based applications. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 456–467. VLDB Endowment, 2004.

[2] L. Aniello, R. Baldoni, and L. Querzoni. Adaptive online scheduling in storm. In *Proceedings of the 7th ACM International Conference on Distributed Event-based Systems*, DEBS '13, pages 207–218, 2013.

[3] V. Cardellini, V. Grassi, F. Lo Presti, and M. Nardelli. Optimal operator placement for distributed stream processing applications. In *Proceedings of the 10th ACM International Conference on Distributed and Event-Based Systems, DEBS 2016*, pages 69–80, 2016.

[4] R. Castro Fernandez, M. Migliavacca, E. Kalyvianaki, and P. Pietzuch. Integrating scale out and fault tolerance in stream processing using operator state management. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 725–736, 2013.

[5] R. Dwarakanath, B. Koldehofe, Y. Bharadwaj, T. A. B. Nguyen, D. Eyers, and R. Steinmetz. Trustcep: Adopting a trust-based approach for distributed complex event processing. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pages 30–39, 2017.

[6] A. Frömmgen, S. Haas, M. Stein, R. Rehner, A. Buchmann, and M. Mühlhäuser. Always the best: Executing transitions between search overlays. In *Proceedings of the 2015 European Conference on Software Architecture Workshops*, ECSAW '15, pages 8:1–8:4, 2015.

[7] A. Frömmgen, B. Richerzhagen, R. Julius, D. Hausheer, R. Steinmetz, and A. Buchmann. Towards the Description and Execution of Transitions in Networked Systems. In *Intelligent Mechanisms for Network Configuration and Security*, pages 17–29. Springer International Publishing, 2015.

[8] T. Heinze, Z. Jerzak, G. Hackenbroich, and C. Fetzer. Latency-aware elastic scaling for distributed data stream processing systems. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS '14, pages 13–22, 2014.

[9] T. Heinze, M. Zia, R. Krahn, Z. Jerzak, and C. Fetzer. An adaptive replication scheme for elastic data stream processing systems. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, DEBS '15, pages 150–161, 2015.

[10] C. Jun and C. Chi. Design of complex event-processing ids in internet of things. In *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, pages 226–229, 2014.

[11] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[12] G. G. Koch, B. Koldehofe, and K. Rothermel. Cordies: expressive event correlation in distributed systems. In *Proceedings of the 4th ACM International Conference on Distributed Event-Based Systems (DEBS 2010)*, pages 26–37, 2010.

[13] B. Koldehofe, R. Mayer, U. Ramachandran, K. Rothermel, and M. Völz. Rollback-recovery without checkpoints in distributed event processing systems. In *Proceedings of the 7th ACM international conference on Distributed event-based systems (DEBS '13)*, page 27, 2013.

[14] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja. Twitter Heron. In *Proceedings of the ACM SIGMOD International Conference on Management of Data - SIGMOD '15*, 2015.

[15] G. T. Lakshmanan, Y. Li, and R. Strom. Placement strategies for internet-scale data stream systems. *IEEE Internet Computing*, 12(6):50–60, 2008.

[16] M. Luthra, B. Koldehofe, and R. Steinmetz. Adaptive Complex Event Processing over Fog-Cloud Infrastructure Supporting Transitions. In *GI/ITG KuVS-Fachgespraech Fog-Computing*, 2018.

[17] M. Luthra, B. Koldehofe, P. Weisenburger, G. Salvaneschi, and R. Arif. TCEP: Adaptive to Dynamic User Environments by Enabling Transitions between Operator Placement Mechanisms. In *Proceedings of 12th ACM International Conference on Distributed and Event-based Systems (DEBS '18)*, 2018.

[18] T. D. Matteis and G. Mencagli. Proactive elasticity and energy awareness in data stream processing. *Journal of Systems and Software*, 127:302 – 319, 2017.

[19] R. Mayer, B. Koldehofe, and K. Rothermel. Predictable low-latency event detection with parallel complex event processing. *IEEE Internet of Things Journal*, 2(4):274–286, 2015.

[20] D. O'Keeffe, T. Salonidis, and P. Pietzuch. Network-aware stream query processing in mobile ad-hoc networks. In *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pages 1335–1340, 2015.

[21] B. Ottenwälder, B. Koldehofe, K. Rothermel, K. Hong, D. Lillethun, and U. Ramachandran. MCEP: A Mobility-Aware Complex Event Processing System. *ACM Transactions on Internet Technology (TOIT)*, 14(1):1–24, 2014.

[22] P. Pietzuch, J. Ledlie, J. Shneidman, M. Roussopoulos, M. Welsh, and M. Seltzer. Network-aware operator placement for stream-processing systems. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 49–49, 2006.

[23] B. Richerzhagen, B. Koldehofe, and R. Steinmetz. Immense Dynamism. *German Research*, 37:24–27, 2015.

[24] B. Richerzhagen, N. Richerzhagen, J. Zobel, S. Schönherr, B. Koldehofe, and R. Steinmetz. Seamless transitions between filter schemes for location-based mobile applications. In *IEEE 41st Conference on Local Computer Networks (LCN)*, pages 348–356, 2016.

[25] B. Richerzhagen, M. Schiller, M. Lehn, D. Lapiner, and R. Steinmetz. Transition-enabled event dissemination for pervasive mobile multiplayer games. In *IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, 2015.

[26] B. Richerzhagen, S. Wilk, J. Rückert, D. Stohr, and W. Effelsberg. Transitions in live video streaming services. In *Proceedings of ACM VideoNEXT 2014*, 2014.

[27] N. Richerzhagen, P. Lieser, B. Richerzhagen, B. Koldehofe, I. Stavrakakis, and R. Steinmetz. Change as chance: Transition-enabled monitoring for dynamic networks and environments. In *14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2018.

[28] S. Rizou, F. Dürr, and K. Rothermel. Solving the multi-operator placement problem in large-scale operator networks. In *Proceedings of 19th International Conference on Computer Communications and Networks*, pages 1–6, 2010.

[29] B. Schilling, B. Koldehofe, and K. Rothermel. Efficient and distributed rule placement in heavy constraint-driven event systems. In *Proceedings of the 13th IEEE International Conference on High Performance Computing and Communications (HPCC-2011)*, pages 355–364, 2011.

[30] F. Starks and T. P. Plagemann. Operator placement for efficient distributed complex event processing in manets. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 83–90, 2015.

[31] T. M. Sutherland, B. Pielech, Y. Zhu, L. Ding, and E. A. Rundensteiner. An adaptive multi-objective scheduling selection framework for continuous query processing. In *9th International Database Engineering Application Symposium (IDEAS'05)*, pages 445–454, 2005.

[32] P. Weisenburger, M. Luthra, B. Koldehofe, and G. Salvaneschi. Quality-aware runtime adaptation in complex event processing. In *Proceedings of the 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '17, pages 140–151, 2017.

[33] Y. Zhang, D. Raychadhuri, B. Dei, E. Baccelli, G. Wang, and B. Ahlgren. Design Considerations for Applying ICN to IoT. *ICN Research Group Internet-Draft*, pages 1–50, 2017.