System Research и проекты НТИ

Vasily A. Sartakov

ksys labs

Agenda

- Обо мне
- НТИ
- История 1: Доверие и отказоустойчивость
- Новые практики
- Hot topics

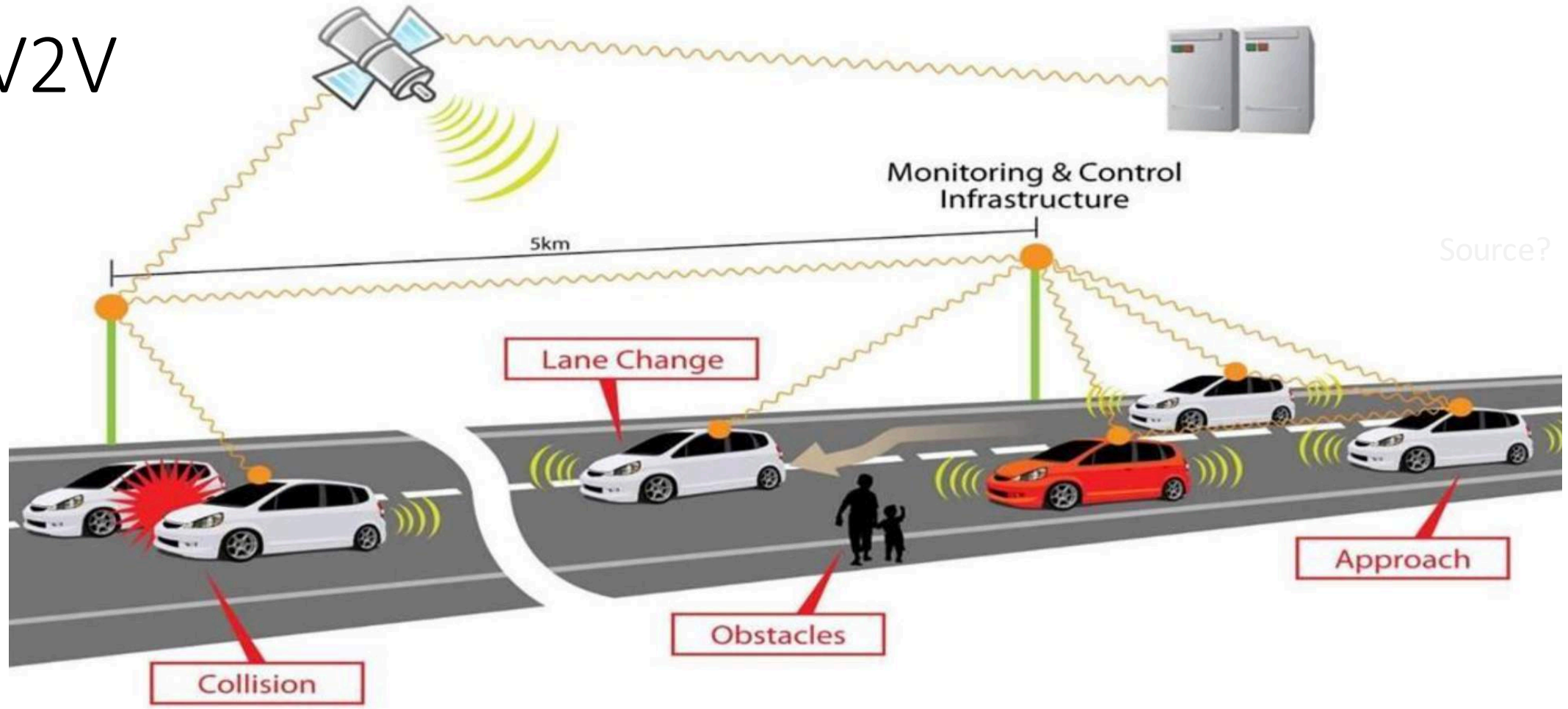
About

- 2004-2007 EE&DSP «Акагео»
- 2007-2008, разработка сетевого оборудования, Лайтком
- 2008-2009, участник команды Montavista, RTSoft
- 2009-2011, разработка прошивок для Ebooks, ст-п Ebookapplications
- 2011-нв, развитие RnD, ksys labs
  - Микроядра
  - IDS
  - Open Source
- 2013-нв, TU Braunschweig, Externer Doktorant
  - Персистентные системы
- Различные форсайт проекты, в частности Форсайт Флот 2015

НТИ

- Аспекты (Социальный, Экономический, Юридический, Институциональный, Технологический)
- Рынки (EnergyNet, FoodNet, SafeNet, HealthNet, AeroNet, MariNet, AutoNet, FinNet, NeuroNet)
  - Динамически создаваемые сети
  - (Интенсивно) взаимодействующих
  - автономных объектов
- Объекты исследований в Systems Research
  - …directly related or having an impact on the development, design, architecture, deployment, and operation of software and hardware systems. (Eurosys)

V2V



Source?

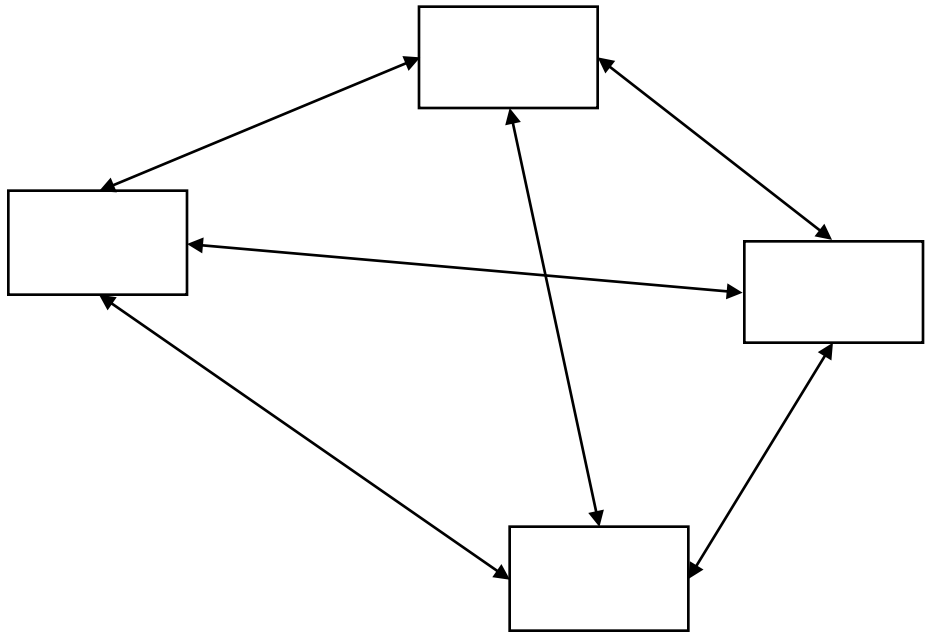
- Распределенную систему
- Операционные системы

- Отказоустойчивость
- Безопасность
- Энергоэффективность

Driverless car, это сложно?

- С 2005 DARPA проводит “Grand Challenge” посвященный Driverless cars.
  - Stanford, MIT, CMU, etc..
  - Совершенно разный стек технологий – от моделей до fuzzy logic
  - Интеграция множества технологий вместе
- Спустя десять лет
  - Любой студент может взять openCV
  - Может собрать конструктор из легко или сделать самому игрушечную платформу
  - И получить в целом как-то двигающийся “Driverless car”
- Достаточно ли этого? Конечно нет
  - Удивительно (на самом деле нет), но в жизни возникают огромное количество исключительных ситуаций

История 1: Доверие и отказоустойчивость

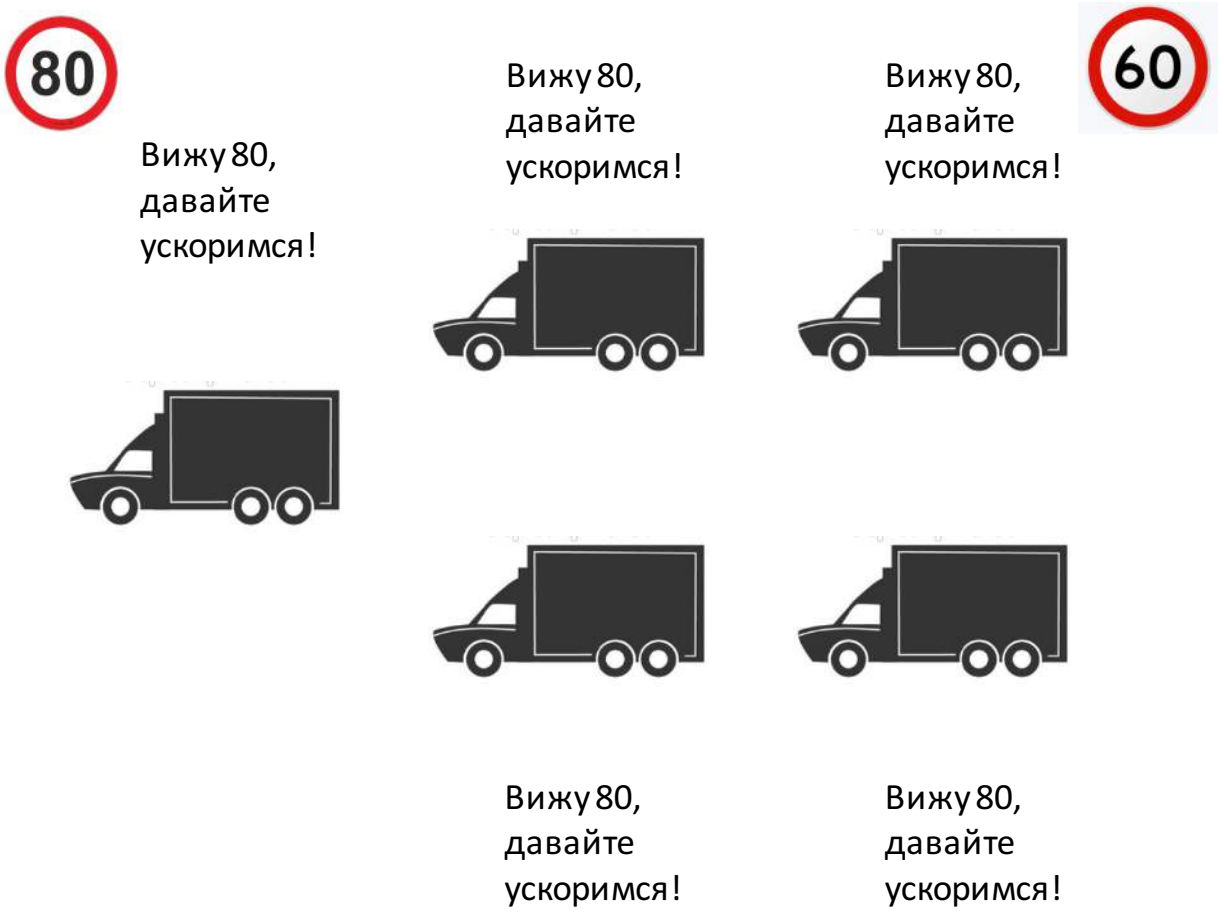


1. Представим себе сеть:
   - Автомобили и инфр.
   - Объекты IoT
   - Дроны

История 1: Доверие и отказоустойчивость

- Автомобильный конвой
  - Постоянная сеть
  - Движение с одной скоростью
  - Каждая машина принимает решение (адепты агентов ликуют)
  - Прежде чем принять решение, нужно посоветоваться с остальными
  - Каждая машина умеет распознавать знаки
- Выезжают на трассу 60->80 км/ч

История 1: Доверие и отказоустойчивость

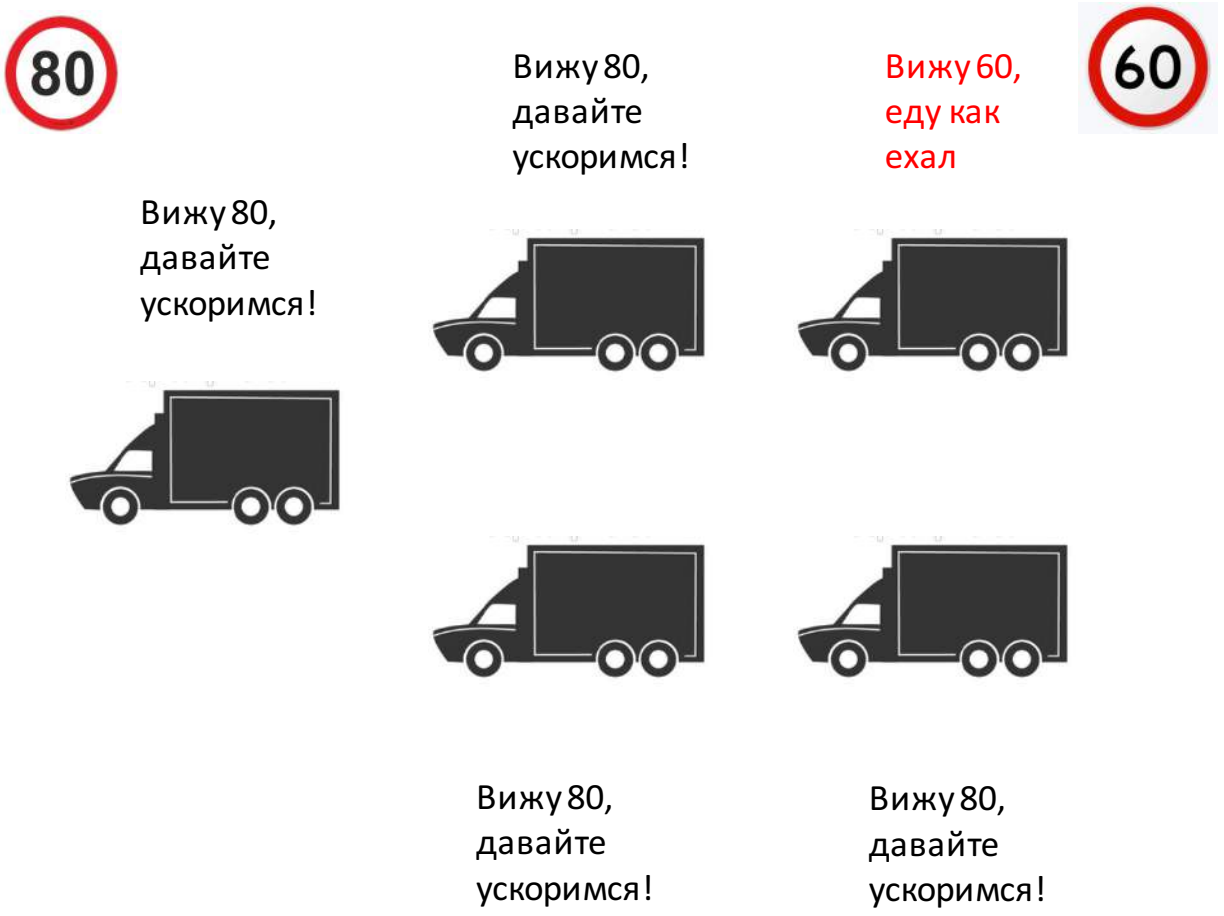


- Автомобильный конвой
  - Постоянная сеть
  - Движение с одной скоростью
  - Каждая машина принимает решение (адепты агентов ликуют)
  - Прежде чем принять решение, нужно посоветоваться с остальными
  - Каждая машина умеет распознавать знаки
- Распознав знак каждая машина отправляет сообщение с целью выработки общего решения (Ускориться до 80 или нет)
  - Все получили сообщение
  - Каким-то образом выработали решение
  - Рассылаем сообщение об ускорении

История 1: Доверие и отказоустойчивость

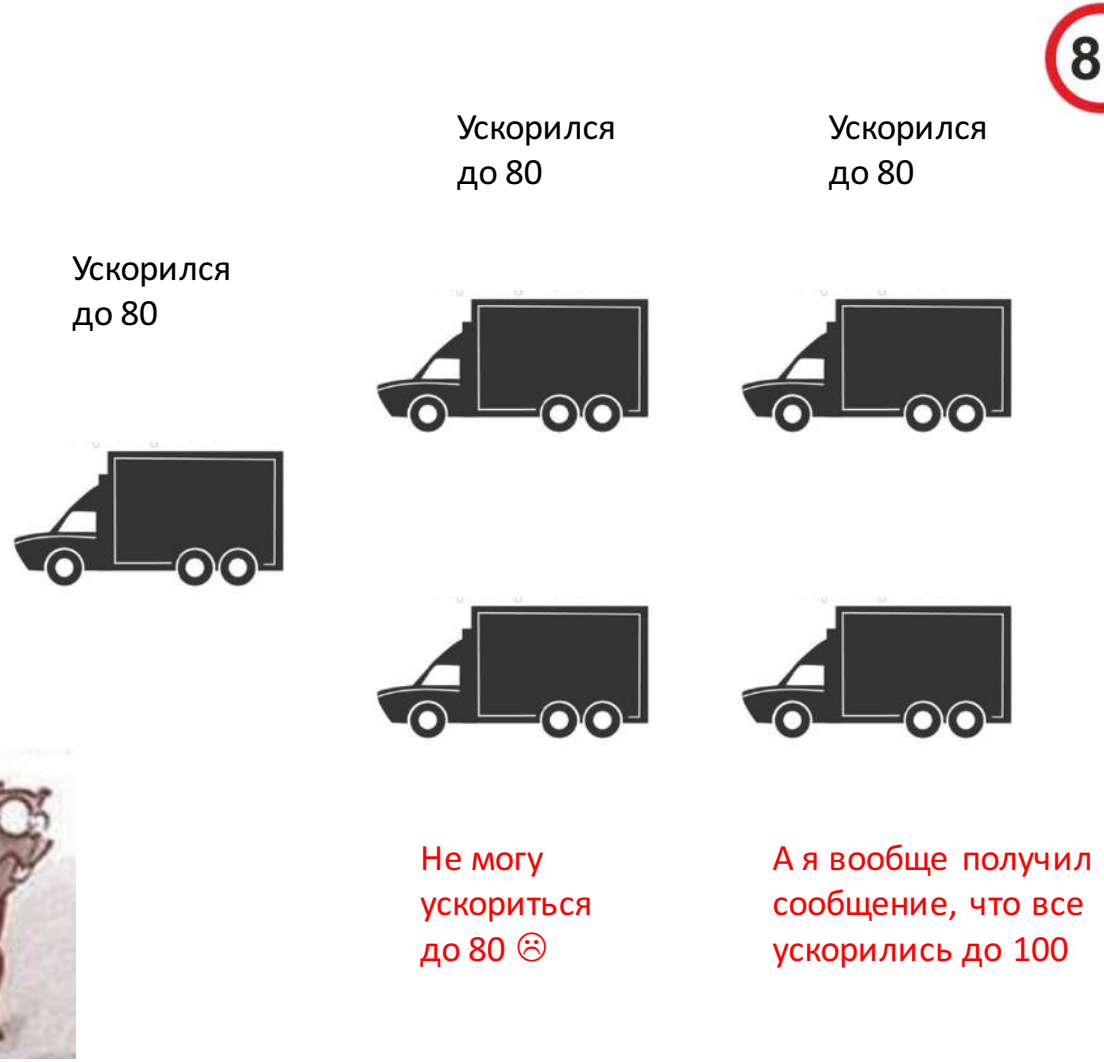
- Автомобильный конвой
  - Постоянная сеть
  - Движение с одной скоростью
  - Каждая машина принимает решение (адепты агентов ликуют)
  - Прежде чем принять решение, нужно посоветоваться с остальными
  - Каждая машина умеет распознавать знаки
- Машины ускорились до 80ти, сообщили, если надо остальным что все ок. Едем дальше. Стоп, а если что-то пошло не так?

История 1: Доверие и отказоустойчивость



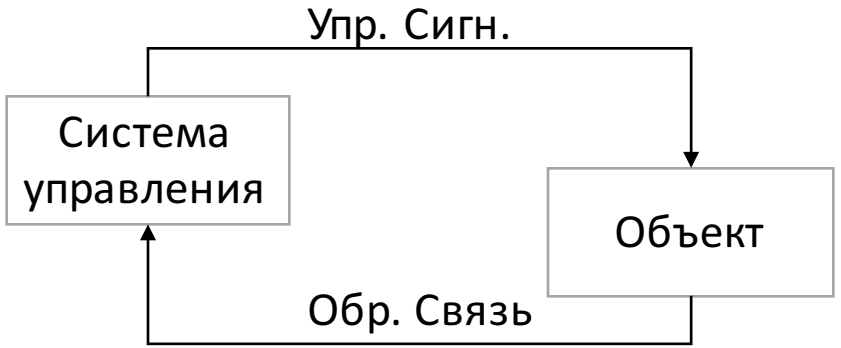
- Автомобильный конвой
  - Постоянная сеть
  - Движение с одной скоростью
  - Каждая машина принимает решение (адепты агентов ликуют)
  - Прежде чем принять решение, нужно посоветоваться с остальными
  - Каждая машина умеет распознавать знаки
- Оказалось, что одна из машин увидела знак 60, а не 80
  - Нет пути назад
  - Нужно принимать решение
- Варианты?
  - Голосовать? 4>1! Если 3?2? 1? 1?
- Хорошо, проголосовали, «продавили»

История 1: Доверие и отказоустойчивость

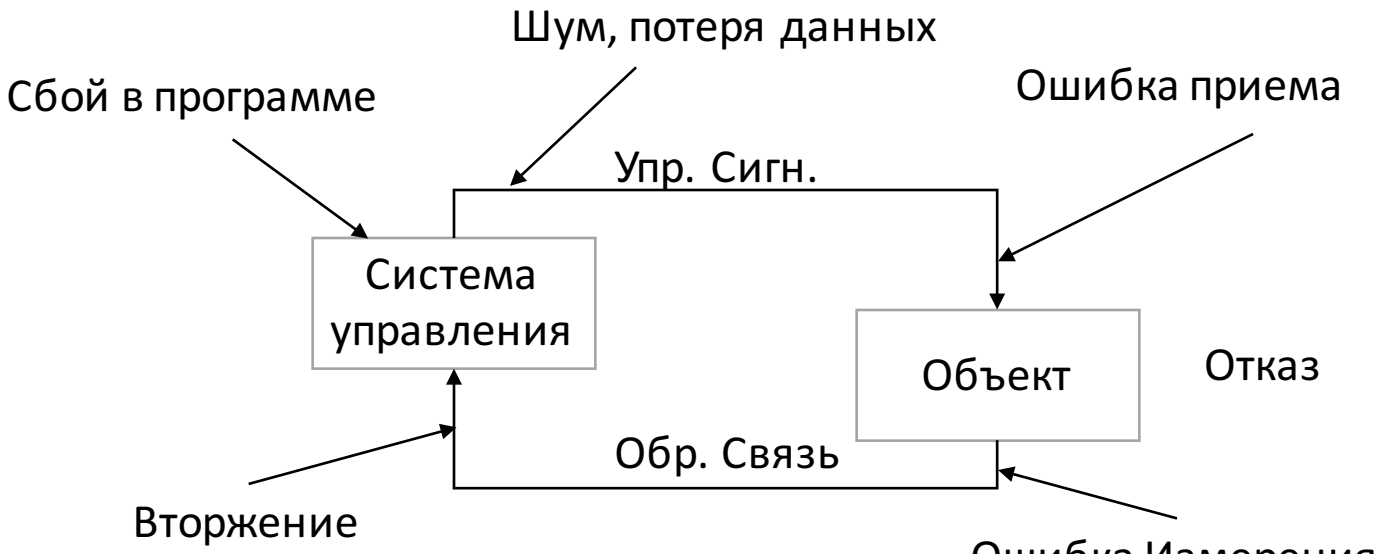


- Автомобильный конвой
  - Постоянная сеть
  - Движение с одной скоростью
  - Каждая машина принимает решение (адепты агентов ликуют)
  - Прежде чем принять решение, нужно посоветоваться с остальными
  - Каждая машина умеет распознавать знаки
- Разослали сообщения, разгоняемся до 80ти, но тут опять что-то пошло не так..
  - Одна из машин не может разогнаться до 80
  - А другая почему-то решила, что разгоняемся до 100

- Многие говорят – автомобили не защищены, я могу их вывести из строя
- Автомобили отвечают – зачем, мы и так себя легко выводим из строя, нам не нужно для этого внешнего вторжения
- Пример выше – попытка поиска консенсуса в распределенной системе. Простой вариант – paxos [1]. Сложный вариант - Byzantine fault tolerance [2,3]
- Все намного сложнее: еще есть пешеходы, погода, наземные устройства, дефекты дороги, ошибки в цифровых моделях, сбои в аппаратуре, сбои в агрегатах, сбои в сбоях(!?)



Кибернетика Здорового человека



Реальное положение дел

Практики

- За последние десятки лет в SR разработано множество технологий (их не достаточно, но есть над чем и с чем работать)
  - Вы ведь не собираетесь брать Linux, как и десятки других стран, которые начали в этом играть еще 10 лет назад?
- Микроядра и окружения (Fiasco.OC [4], L4Re [5], Genode [6], NOVA [7], NRE/NUL [8], seL4[9])
  - Open Source
  - Возможность создавать новые проекты и быть центрами компетенций
  - Возможность включения в существующие исследовательские проекты
- Программное мажорирование
  - Репликация (L4Reanimator, Romain[10])
  - Отказоустойчивые ОС (NewtOS [11, 12])
- Консенсус протоколы
  - Paxos [1], practical BFT [2], CheapBFT [3]
- Верификация
  - seL4 [9]

Верификация как практика

- Какие-то наработки в области верицикации
- seL4 в открывет новые возможность
  - Представьте – мат. модели всего
  - Дрон не залетает в окно и это математически доказано
  - Энергетическое оборудование сертифцируется по модели и прошивке прямо в цифровой модели города
  - В “Нью-Васюках” автомобили патчат свой код и модели налету
    - Подождите, а как же DARPA Cyber Grand Challenge?...
- Внедрение верификации в методологию и практику программирования
- (Системное) Программирование как разновидность высококвалифицированной рабочей профессии
- Нужно больше ~~золота~~ Open Source проектов
- Нужно включаться в SR проекты и существенно менять политику поддержки исследований

Hot and future topics

- Foresight of Systems Research
  - SIG OS VDE
  - Будет опубликован весной 2016
- “Energy efficiency is a new performance”
  - Power-aware systems
  - Масштабируемость при минимальном/постоянном/максимальном потреблении
  - ЭЭ криптография, коммуникации
- Persistent Systems
- Self-aware systems

Проекты и статьи

1. Pease, M., Shostak, R., & Lamport, L. (1980). Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, *27*(2), 228-234.
2. Castro, M., & Liskov, B. (1999, February). Practical Byzantine fault tolerance. In *OSDI* (Vol. 99, pp. 173-186).
3. Kapitza, R., Behl, J., Cachin, C., Distler, T., Kuhnle, S., Mohammadi, S. V., ... & Stengel, K. (2012, April). CheapBFT: resource-efficient byzantine fault tolerance. In *Proceedings of the 7th ACM european conference on Computer Systems* (pp. 295-308). ACM.
4. https://os.inf.tu-dresden.de/fiasco/
5. https://os.inf.tu-dresden.de/L4Re/
6. http://www.genode.org
7. http://hypervisor.org
8. https://github.com/TUD-OS/NUL
9. https://sel4.systems
10. Döbel, B., Muschner, R., & Härtig, H. (2014). Resource-aware replication on heterogeneous multicores: Challenges and opportunities. *arXiv preprint arXiv:1405.2913*.
11. Hruby, T., Bos, H., & Tanenbaum, A. S. Towards Optimal Scheduling of Multiserver System Components.
12. Hruby, T., Vogt, D., Bos, H., & Tanenbaum, A. S. (2012, June). Keep net working-on a dependable and fast networking stack. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on* (pp. 1-12). IEEE.

Спасибо за внимание.

Vasily A. Sartakov
sartakov@ksyslabs.org