Network Calculus Based Approach to DDoS Detection and Mitigation in CPS

William Emfinger

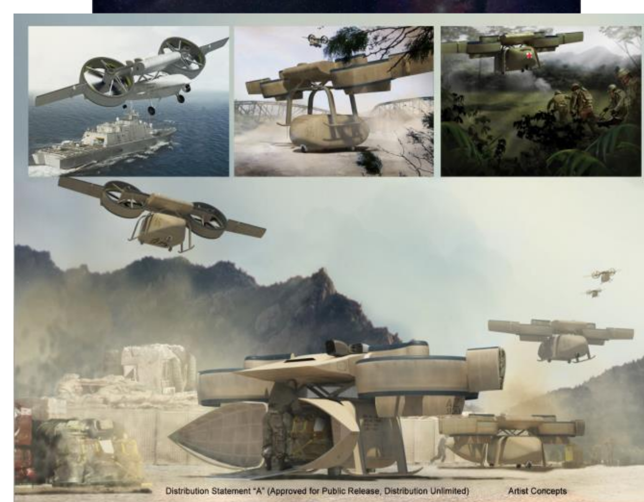
04-11-2015

Outline

* Introduction
* Network Calculus fundamentals
* Network Performance Analysis for CPS Applications : PNP²
* PNP² DDoS Detection
* DDoS Mitigation
* Conclusions

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

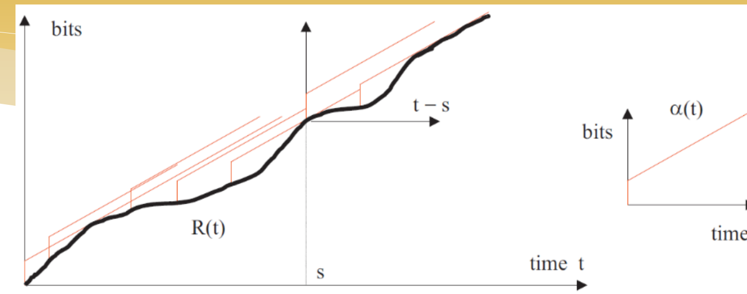
Introduction

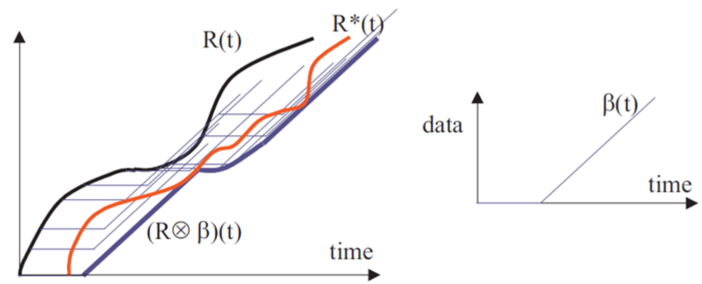


- Cyber-Physical Systems (CPS):
  - According to NIST: *CPS are smart systems that include co-engineered interacting networks of physical and computational components*
  - Embedded computers, sensing/actuation software, tightly coupled physical system
  - Communications network (wireless) acts as backbone for cooperation between nodes
  - Network resources **provided as a service** to users
- Network resource availability affected by both application load and system's environment
  - Bandwidth (bits/sec) required
  - Buffering delay and transmission delay incurred
  - Availability of network resources
  - Some systems may have **deterministic, time-varying network characteristics** and/or application **network load**
- Design-time analysis is critical for applications that require tight and/or real-time guarantees
  - Use the analysis to perform **test for admission** into the system

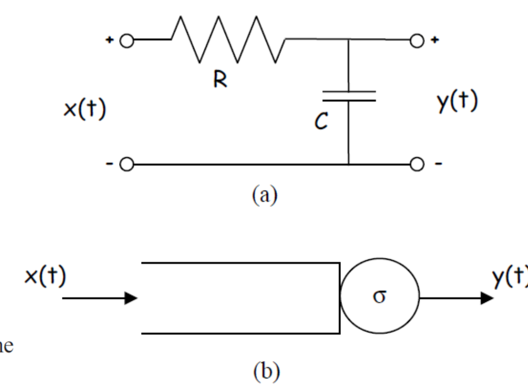
FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

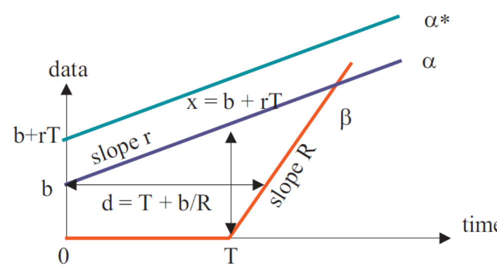
Network Calculus Fundamentals





- Based on (min,+)-calculus
- Abstracts network traffic and computing nodes as *arrival curves* and traffic shaping *service curves*
- Analytically predicts worst-case buffer requirements and application buffering delay
- Applies system-theory concepts to computer network analysis
  - Input function (data input flow) convolved with a system function (shaper) to produce the output function (data output flow)
- Allows the calculation of deterministic performance bounds on
  - Buffering delay
  - Buffer backlog

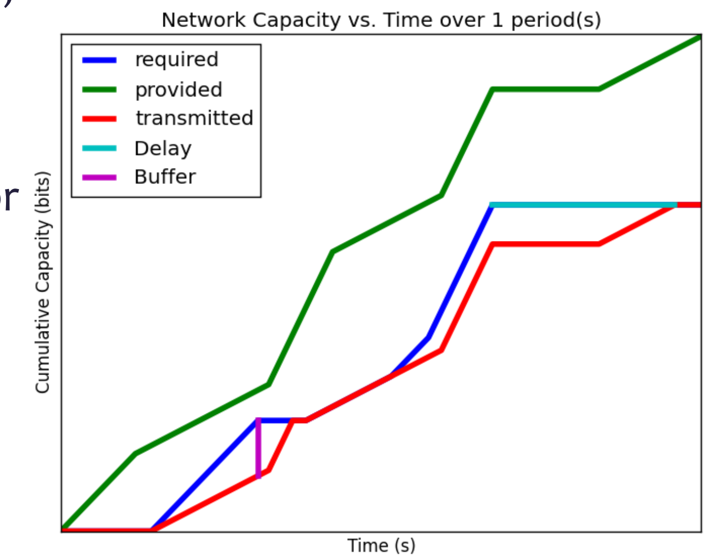




FORCES
FOUNDATIONS OF RESILIENT
CYBER-PHYSICAL SYSTEMS

Network Performance Analysis for CPS Applications

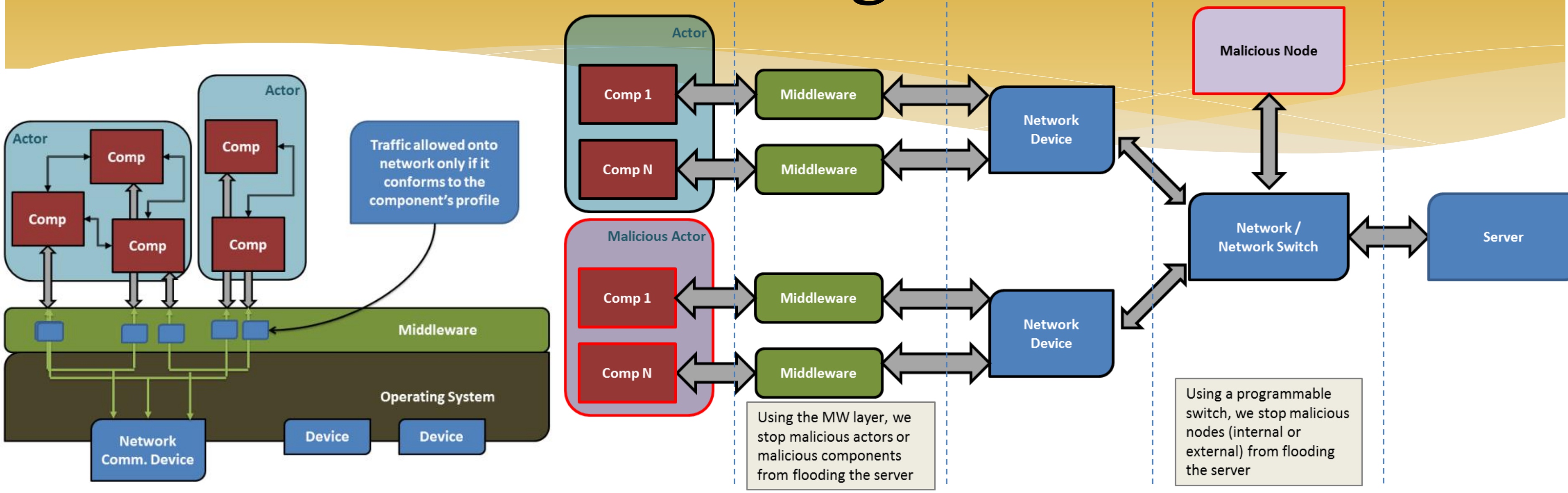
- To address the issues described above, we have developed a network modeling paradigm similar to Network Calculus' arrival curves and service curves, called ***P**recise **N**etwork **P**erformance **P**rediction* (**PNP²**)
- In contrast to Network Calculus, we precisely model the network traffic as a function of time, i.e. bits produced or serviced as functions of time – called *profiles*
- Convolve the application data profile with the network capacity profile to obtain transmitted data profile
  - *r[t]* : application (*required*) data profile
  - *p[t]* : network (*provided*) capacity profile
  - *o[t]* : the transmitted (*output*) data profile
- Assumptions:
  - Know at design-time how the application and the network is expected perform
  - Time-Synchronization between the nodes
  - State-less transmission protocols, e.g. UDP
  - No packet loss or transmission errors



$$y = o[t] = (r \otimes p)[t]$$
$$= \min\big(r[t], p[t] - (p[t-1] - o[t-1])\big)$$
$$buffer = \sup\{r[t] - o[t] \;:\; t \in \mathbb{N}\}$$
$$delay = \sup\{o^{-1}[y] - r^{-1}[y]\}$$

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

DDoS Mitigation



- Using PNP²:
  - Precisely model application network behavior, with some bounds on deviation
- Assume infrastructure is controlled and verified
  - only applications may be compromised
- Middleware detects that application traffic production deviates from model
- Use out-of-band communication between server and clients
  - Server sees multiple clients simultaneously producing more data than normal
  - Informs client-side middleware to throttle clients and prevent denial of service

Conclusions

* We have developed a foundation for precisely modeling and analyzing at design-time applications and CPS networks
* Implementing the modeling semantics in a trusted middleware allows management of the network performance of the CPS at run-time
* Since these models capture the behavior of well-behaved applications, the middleware can detect deviations from this behavior and act accordingly
  * Sender side detection: data produced far exceeds the model; throttle
  * Receiver side detection: each sender produces slightly more data than the model and is allowed on the network; receiver cannot handle aggregate traffic so must discriminate good senders from bad
* Using capabilities in the middleware, the misbehaving senders can be throttled at the source

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Thank you!

Questions?

Berkeley UNIVERSITY OF CALIFORNIA

Massachusetts Institute of Technology

NSF

MICHIGAN

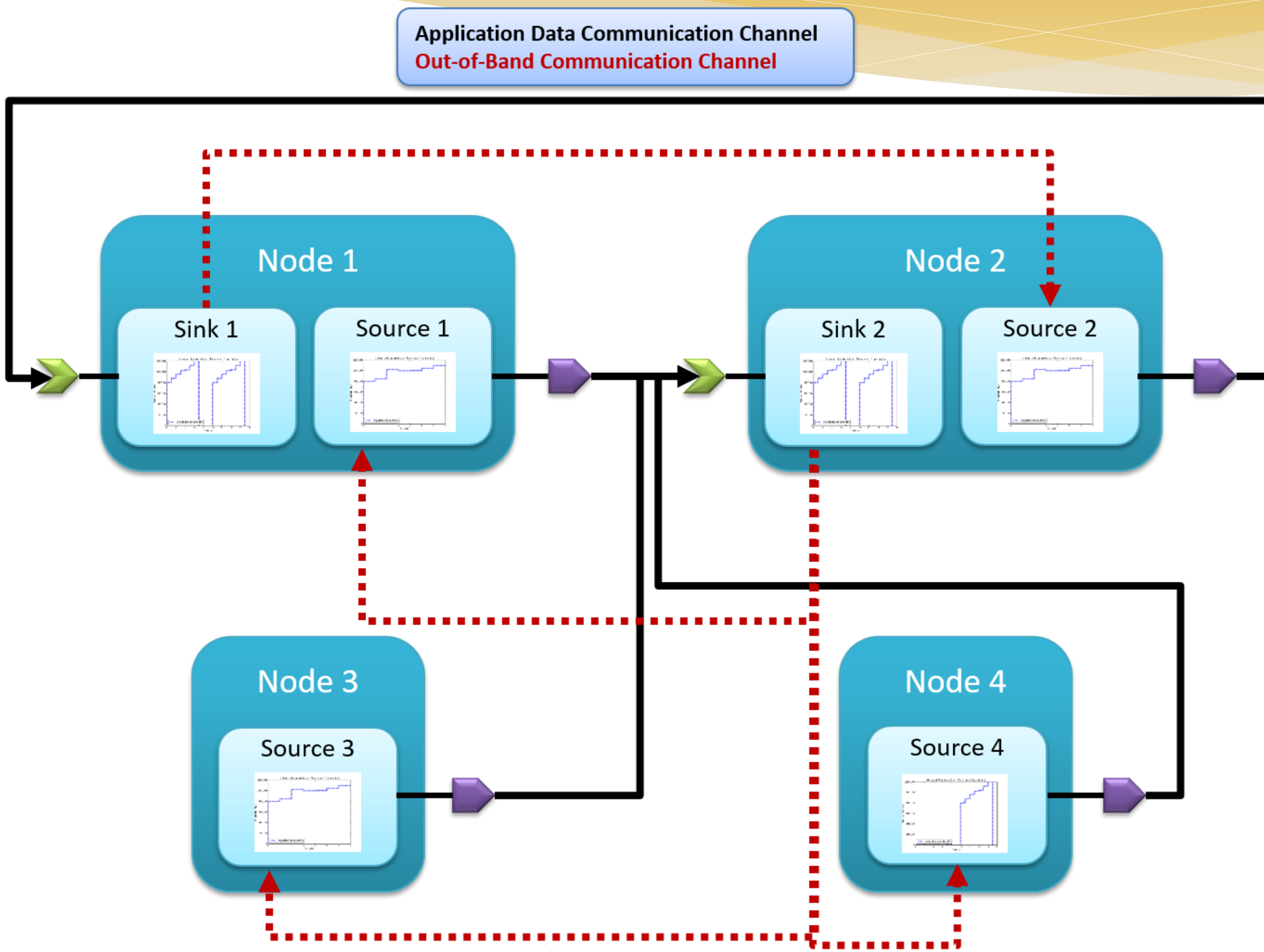
VANDERBILT UNIVERSITY

Algorithm

```
receiver::limit_ddos( t_start, t_end )
{
  for sender in senders
  {
    d_start = sender.received_data(t_start)
    d_end = sender.received_data(t_end)
    profile_d_start =
      sender.profile(t_start)
    profile_d_end =
      sender.profile(t_end)
    allowed_data = profile_d_end - profile_d_start
    actual_data = d_end - d_start
    if actual_data > allowed_data
    {
      sender.disable()
    }
  }
}
```


FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Example



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

PNP^2 DDoS Detection

* DDoS attacks are generally classified as excessive traffic from a large amount of (possibly heterogeneous) sources targeted towards a single point or a single group
* The receiver has a finite receive buffer and a specified profile governing the rate at which it will pull data from the buffer:
  * Senders who send too much data may cause the buffer to fill up, causing data from well-behaved senders to be lost
  * If we can detect which senders are misbehaving and prevent them from sending, well-behaved senders will not see any degradation of service
* Developed an out-of-band (OOB) communications channel from the sender middleware to the receiver middleware:
  * Invisible to the application
  * Allows the receiver to communicate with and control the sender's middleware
* Works because we know precisely the correct behavior of applications and we know all of the communication patterns and we have a trusted middleware

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS