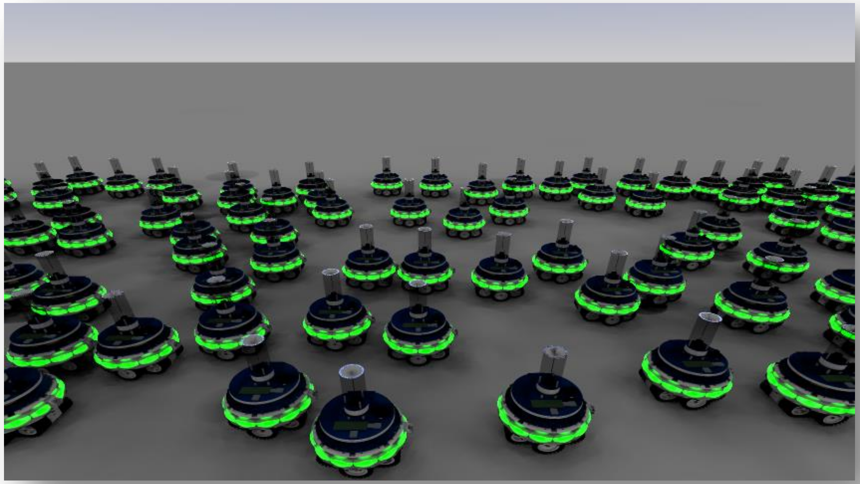
A Distributed Consensus Protocol for Clock Synchronization in Wireless Sensor Networks

Luca Schenato
Giovanni Gamba

Presented by:
Matt Kwan

Tufts

Relevance



Objective

- Bring all virtual clocks to a **consensus** via a **linear function** of the nodes' own **local clocks**.

Average TimeSync Protocol

Problem Formulation

$\tau_i(t) = \alpha_i t + \beta_i$ (1) *Absolute time representation.*

where τ_i represents the local time of node i, α_i represents the rate of node i, and β_i represents the offset of node i.

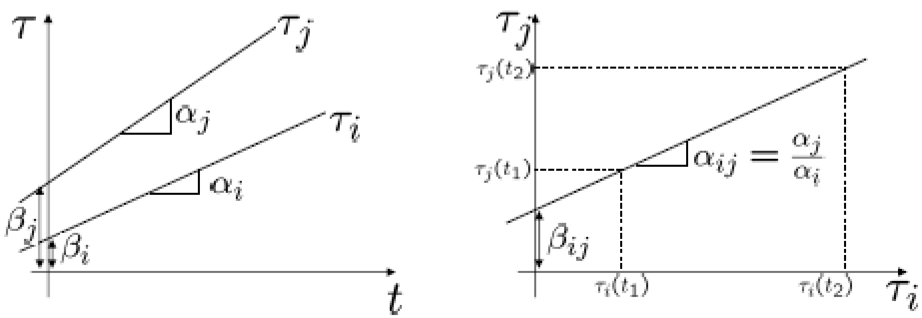


Figure taken from [1].

$\hat{\tau}_i(t) = \hat{\alpha}_i \tau_i + \hat{o}_i$ (2) *Virtual clock estimation as a linear function.*

where $\hat{\alpha}_i$ represents virtual clock skew estimate and $\hat{o}_i$ represents virtual clock offset estimate.

Solution – The ATS Protocol

1. **Relative Skew Estimation**
2. Skew Compensation
3. Offset Compensation

- We need to reference our linked neighbors to decide how to adjust.

$$\eta_{ij}^{+} = \rho_n \eta_{ij} + (1 - \rho_n) \frac{\tau_j(t_2) - \tau_j(t_1)}{\tau_i(t_2) - \tau_i(t_1)} \quad (3)$$

where η_{ij}^{+} is the next value of the relative skew *estimate* between nodes *i* and *j*, and ρ_n is a tuning factor (speed vs. noise-protection).

This estimate converges to the *true* relative skew:

$$\eta_{ij}(t_k) = \rho_\eta^k \eta(0) + \sum_{l=1}^{k-1} (1 - \rho_\eta)^l \alpha_{ij} = \rho_\eta^k \eta(0) + \alpha_{ij}(1 - \rho_\eta^k) \quad (4)$$

Taking the limit as *k* approaches infinity gives us

$$\lim_{k \to \infty} \eta_{ij}(t_k) = \alpha_{ij} \quad (5)$$

Solution – The ATS Protocol

1. Relative Skew Estimation
2. **Skew Compensation**
3. Offset Compensation

- Given the relative skew between nodes, we can estimate a relative skew between a node and the virtual clock.

$$\hat{\alpha}_i^{\,+} = \rho_v \hat{\alpha}_i + (1 - \rho_v)\eta_{ij}\hat{\alpha}_j \qquad (5)$$

where $\hat{\alpha}_i$ is the estimated virtual skew of i and ρ_v is a tuning factor.

This can be rewritten as:

$$x_i^{\,+} = \rho_v x_i + (1 - \rho_v) x_j \qquad (6)$$

which can be taken in matrix form for all the nodes as:

$$\mathbf{x}^+ = A\mathbf{x} \qquad (7)$$

where $A \in \mathbb{R}^{nxn}$ has ones on the diagonal and zeros elsewhere except the i-th row where $A_{ij} = 1 - \rho_v$ and $A_{ii} = \rho_v$.

Tufts UNIVERSITY

Solution – The ATS Protocol

1. Relative Skew Estimation
2. Skew Compensation
3. **Offset Compensation**

- Now that the speeds are corrected, we only have to adjust all the offsets.

$$\begin{aligned} \hat{o}_i^+ &= \hat{o}_i + (1-\rho_o)(\hat{\tau}_j - \hat{\tau}_i) \\ &= \hat{o}_i + (1-\rho_o)(\hat{\alpha}_j\tau_j + \hat{o}_j - \hat{\alpha}_i\tau_i - \hat{o}_i) \end{aligned} \tag{8}$$

where o_i represents virtual offset, $\hat{\alpha}_i$ is the estimated virtual skew of i, and ρ_o is a tuning factor.

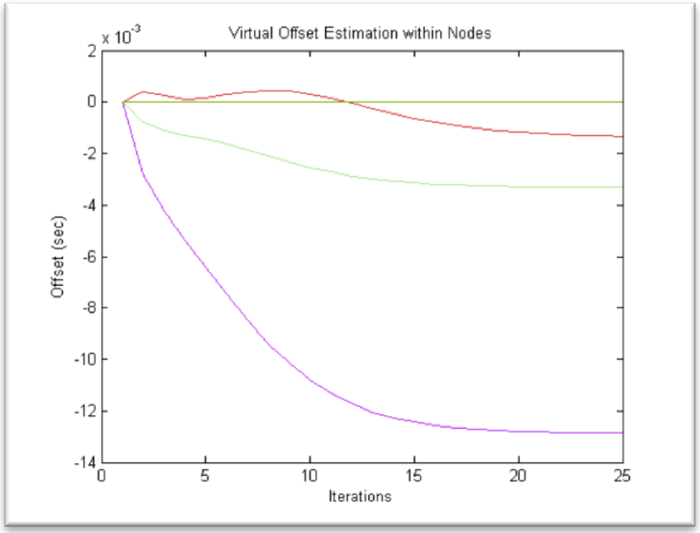
As long as there is a connected path from any node i to any other node j, this limit will hold:

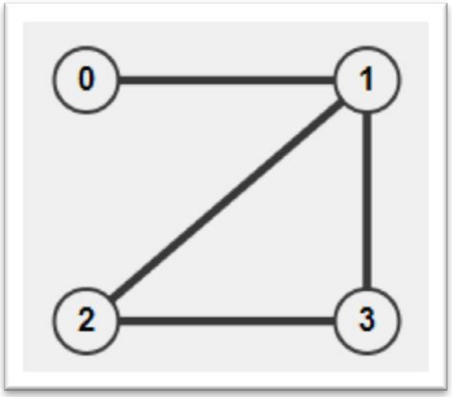
$$\lim_{t\to\infty} \hat{\tau}_i(t) = \hat{\tau}_j(t), \quad \forall (i,j) \tag{9}$$

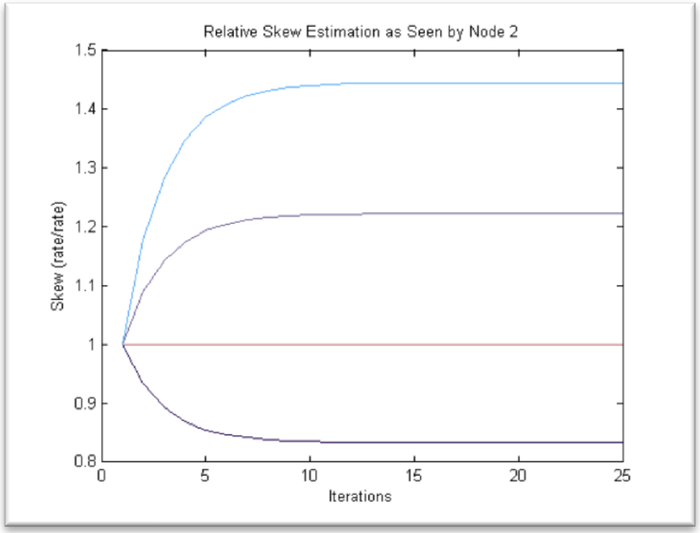
which means the estimations will converge to a single virtual time.

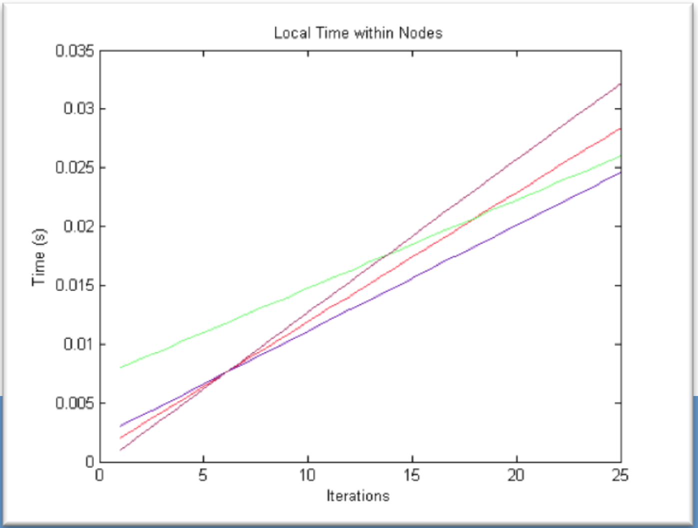
Tufts UNIVERSITY

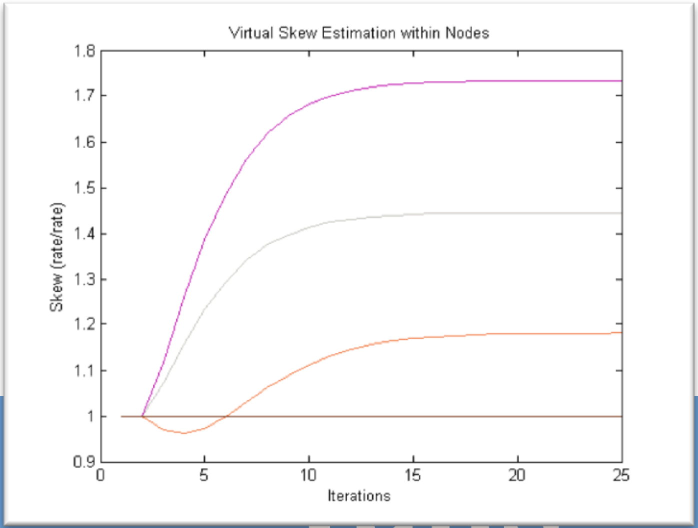
Results – 4 nodes (1/4)





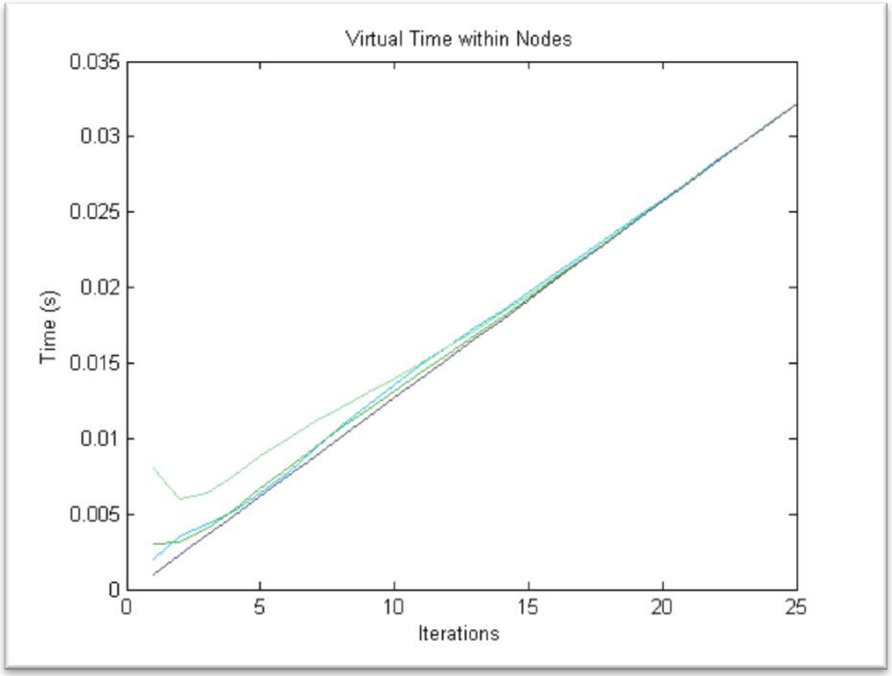


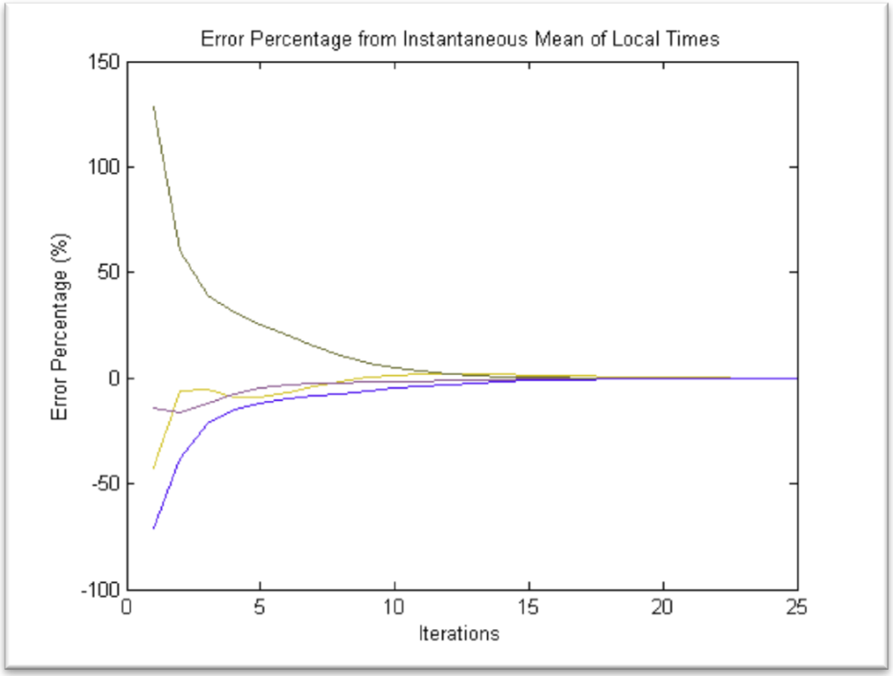




UNIVERSITY

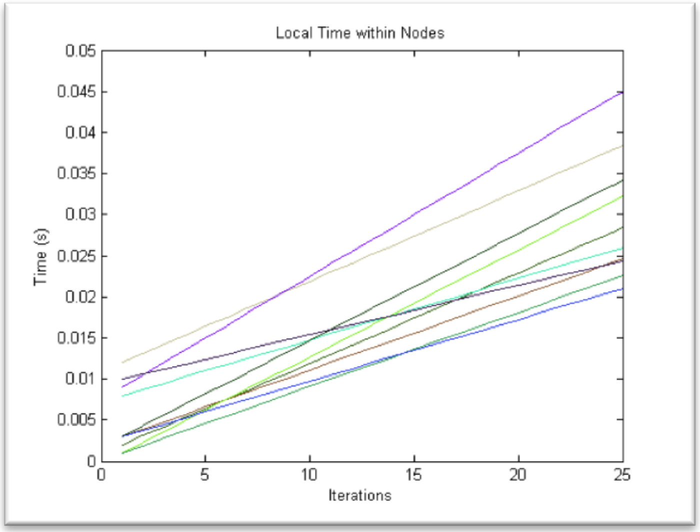
Results – 4 nodes (2/4)

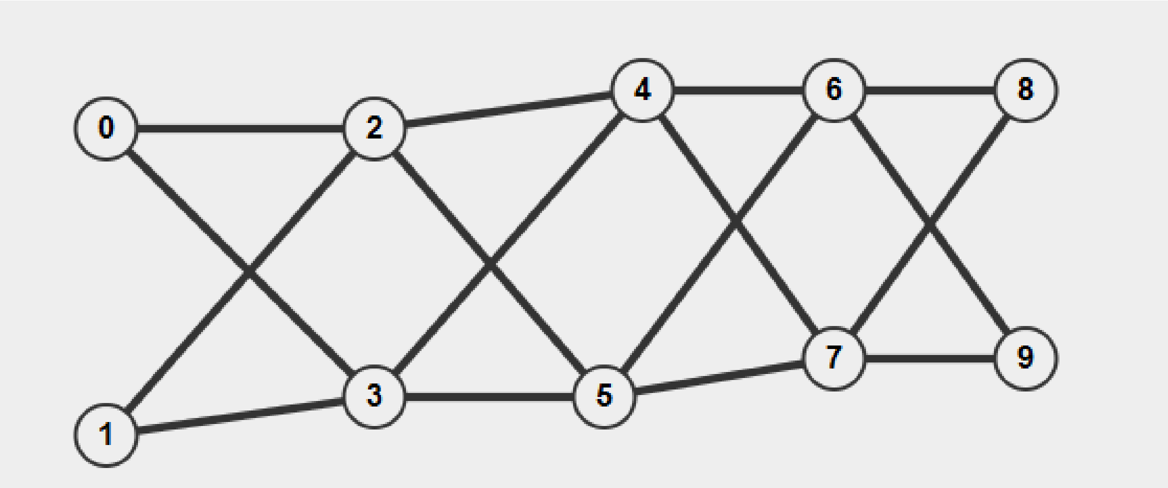


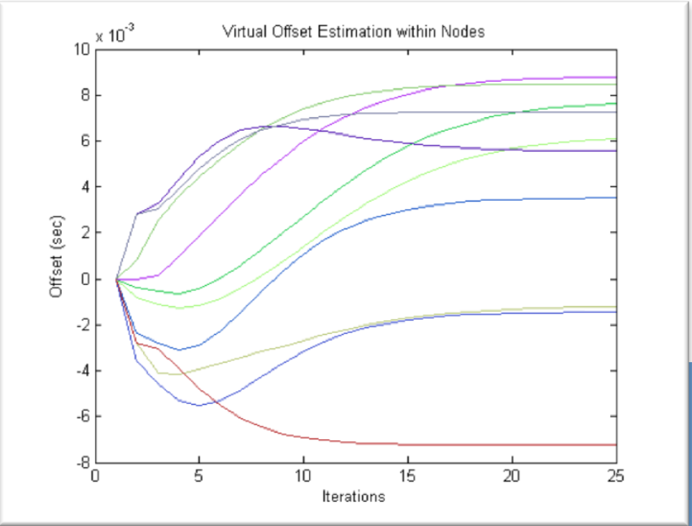


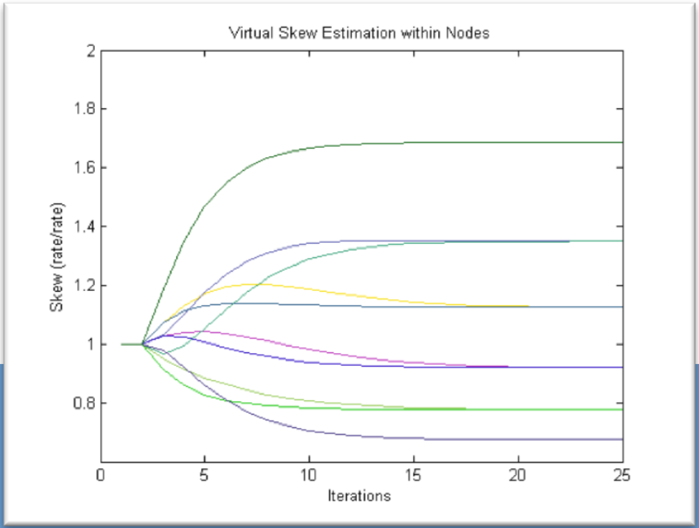
Tufts UNIVERSITY

Results – 10 nodes (3/4)

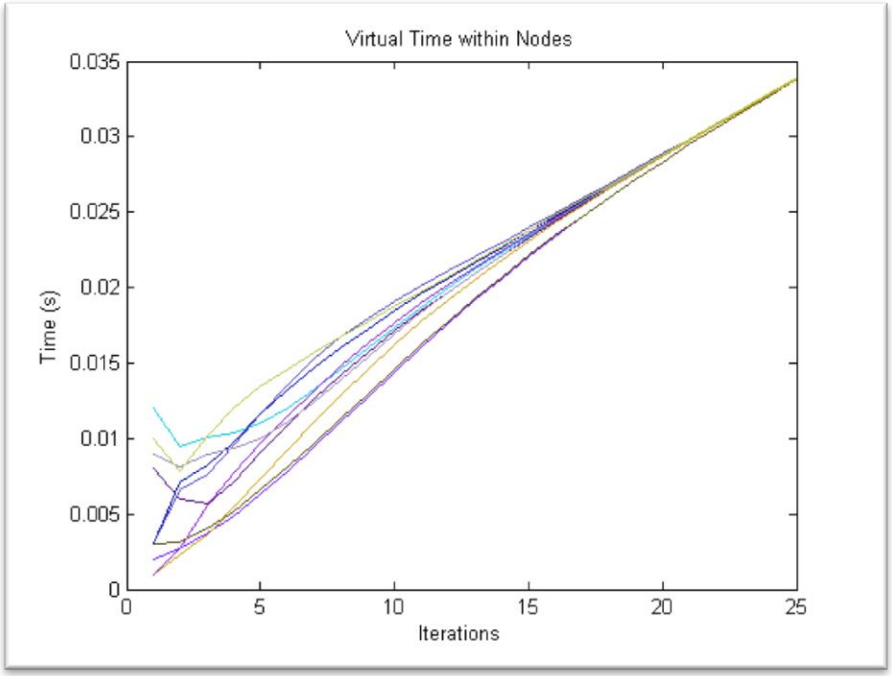


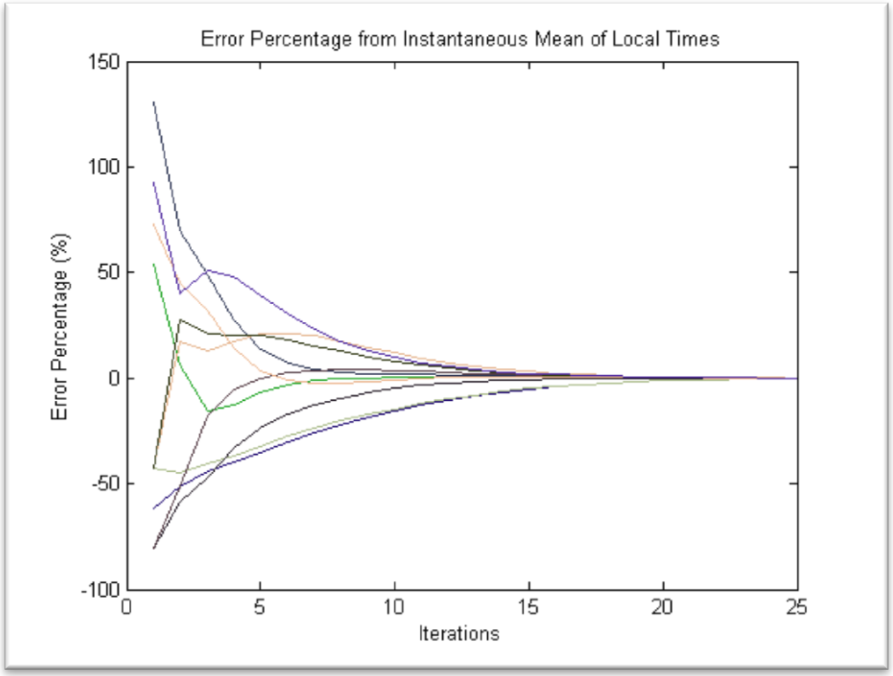






Results – 10 nodes (4/4)





Thank you!
Questions?

Tufts UNIVERSITY